IBM File Manager for z/OS

User's Guide and Reference

Version 13 Release 1

SC19-4119-00

Note
Before using this document, read the general information under "Notices" on page 1219.

First Edition (October 2013)

This edition applies to Version 13 Release 1 Modification Level 0 of IBM File Manager for z/OS (program number 5655-Q12) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. For information on how to send comments, see "How to send your comments to IBM" on page xv.

This publication is available on the Web at:

www.ibm.com/software/awdtools/filemanager/

Contents

About this document

This document provides guidance and reference information for users of IBM® File Manager for z/OS® working with QSAM data sets, VSAM data sets, IAM data sets, PDS members, Websphere MQ queues, or HFS files. (Throughout this document, the z/OS UNIX file systems, HFS and zFS, are referred to as "HFS".)

File Manager for z/OS contains four “flavors” of File Manager, for working in different environments or with different data formats:

- **File Manager** “base” component (described in this document), for working with QSAM data sets, VSAM data sets, IAM data sets, PDS members, Websphere MQ queues, CICS® resource, or HFS files as an ISPF application or by means of the File Manager Plug-in for Eclipse.
- **File Manager DB2® component**, for working with DB2 data sets
- **File Manager IMS™ component**, for working with IMSdata sets
- **FM/CICS** for working with CICS resources.

This document describes the File Manager Base component. For information about the DB2, IMS, and CICS components, see these documents:

- *File Manager User's Guide and Reference for DB2 Data*, SC19-4120
- *File Manager User's Guide and Reference for IMS Data*, SC19-4121
- *File Manager User's Guide and Reference for CICS*, SC19-4122

In the rest of this document, the term "File Manager" refers only to the File Manager Base component, not to the DB2, IMS, or CICS components.

This document is divided into three parts:

Part 1. Working with File Manager

Describes the concepts inherent within File Manager, provides step-by-step instructions on how to perform tasks using the File Manager panels under ISPF, and describes how to use the File Manager functions in batch jobs, REXX procedures or TSO CLISTs. The information is presented in a task-based format, with panel, field, command, and function definitions provided only when they are relevant to the current task. For a complete description of a particular panel, field, command or function, you should use the appropriate chapter in **Part 2 File Manager References**.

Users who are new to File Manager should find that Part 1 presents the tasks in logical groupings, helping you to use the guide as a learning tool, as well as enabling you to quickly find the specific task instructions you need.

Part 2. File Manager References

Provides a lookup reference for File Manager panels, commands, and functions. The information is presented in three main chapters:

1. File Manager Panel and Field Reference: Lists all of the File Manager panels, provides a definition for each field in the panel and, where applicable, lists the value ranges that are valid for each field.
2. File Manager Command Reference: Lists all of the Primary and Prefix commands and gives a brief description of its syntax, usage and equivalent function.

3. File Manager Function Reference: Lists all of the File Manager functions and gives a description of its syntax, usage and available parameters.

Both new and experienced users of File Manager can use Part 2 to quickly look up such things as the exact syntax of a command or function, or the acceptable values for a field in a panel.

Part 3. Appendixes

The Appendix is made up of the File Manager ("base") messages, and provides the full text explanation of all FM "base" batch error messages.

The support information tells you about IBM Web sites that can help you answer questions and solve problems, followed by notices, bibliography, glossary and index.

Who should use this document

This document is for three kinds of File Manager users:

- Application programmers who need to test and debug programs
- Service support personnel who need to analyze and fix problems
- System administrators and system operators who need to do routine tasks such as moving large amounts of data

This document assumes that you are familiar with using ISPF.

To use File Manager functions in batch jobs, you must already be familiar with JCL.

To use File Manager functions in REXX procedures, you must already be familiar with the REXX programming language.

Prerequisite and related information

File Manager documentation supports the following tasks:

Evaluating File Manager

File Manager Fact Sheet provides an overview of the product to help the business professional decide if File Manager has the functions needed.

Planning for, installing, customizing, and maintaining File Manager

Refer first to the *File Manager Customization Guide* which may then refer you to the *File Manager Program Directory* (shipped with the product tape) for some information.

Using File Manager

This document, the *File Manager User's Guide and Reference*, is a guide to using File Manager. When using File Manager under ISPF, you can also refer to the online help.

For information about related products, see the "Bibliography" on page 1223.

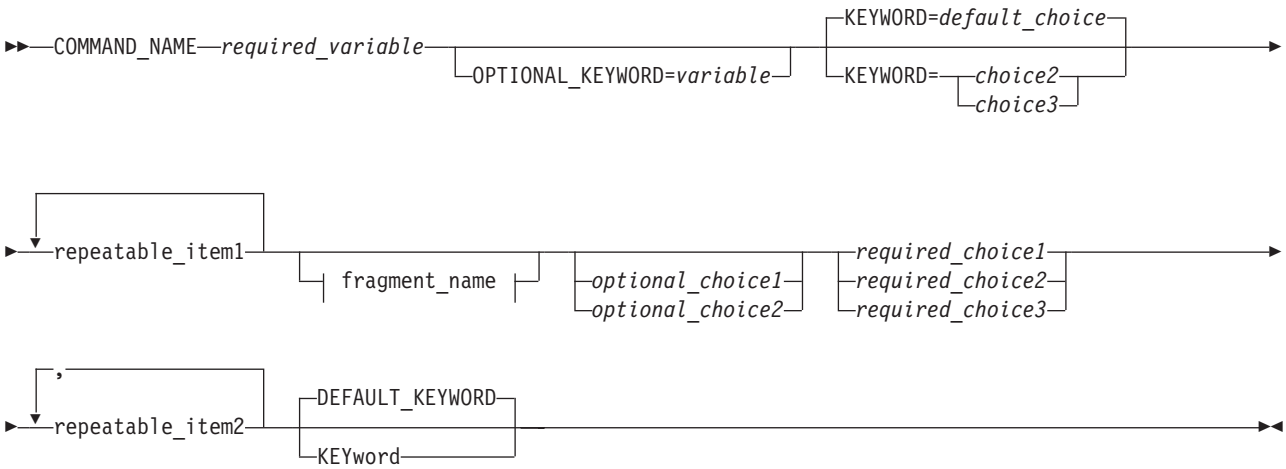
How to read syntax diagrams

The syntactical structure of commands described in this document is shown by means of syntax diagrams.

Figure 1 on page xiii shows a sample syntax diagram that includes the various notations used to indicate such things as whether:

- An item is a keyword or a variable.
- An item is required or optional.
- A choice is available.
- A default applies if you do not specify a value.
- You can repeat an item.

Syntax



fragment_name:

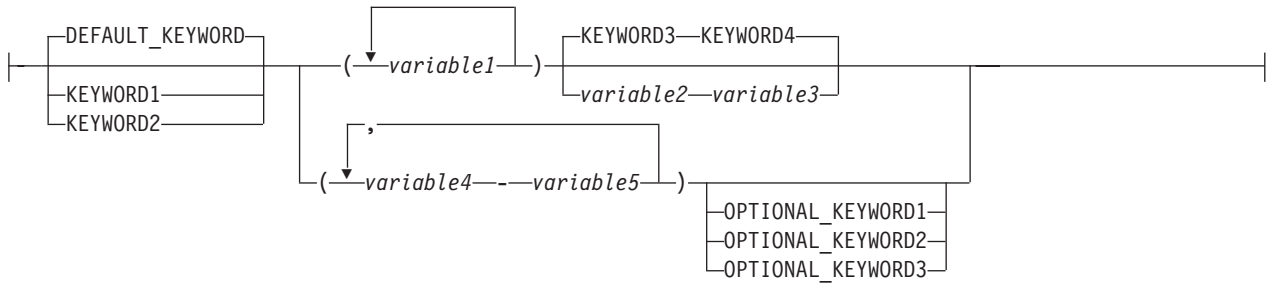


Figure 1. Sample syntax diagram

Here are some tips for reading and understanding syntax diagrams:

Order of reading

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

The ▸▸— symbol indicates the beginning of a statement.

The —▸ symbol indicates that a statement is continued on the next line.

The ▸— symbol indicates that a statement is continued from the previous line.

The —▸◂ symbol indicates the end of a statement.

Keywords

Keywords appear in uppercase letters.

Sometimes you only need to type the first few letters of a keyword, The required part of the keyword appears in uppercase letters.

In this example, you could type "KEY", "KEYW", "KEYWO", "KEYWOR" or "KEYWORD".

The abbreviated or whole keyword you enter must be spelled exactly as shown.

Variables

Variables appear in lowercase letters. They represent user-supplied names or values.

Required items

Required items appear on the horizontal line (the main path).

Optional items

Optional items appear below the main path.

Choice of items

If you can choose from two or more items, they appear vertically, in a stack.

If you *must* choose one of the items, one item of the stack appears on the main path.

If choosing one of the items is optional, the entire stack appears below the main path.

If a default value applies when you do not choose any of the items, the default value appears above the main path.

Repeatable items An arrow returning to the left above the main line indicates an item that can be repeated.

If you need to specify a separator character (such as a comma) between repeatable items, the line with the arrow returning to the left shows the separator character you must specify.

Fragments

Where it makes the syntax diagram easier to read, a section or *fragment* of the syntax is sometimes shown separately.

fragment_name:

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to comments@us.ibm.com
2. Use the form on the Web at:
   www.ibm.com/software/ad/rcf/
3. Mail the comments to the following address:

   IBM Corporation
   H150/090
   555 Bailey Avenue

San Jose, CA
95141-1003
U.S.A.

Include the following information:
- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  FM V13R1 Customization Guide
  SC19-4118-00
- The topic and page number related to your comment
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:
- Contact your IBM service representative
- Call IBM technical support
- Visit the IBM support portal at http://www.ibm.com/systems/z/support/

Summary of changes

This edition of the document provides information applicable to File Manager Version 13 Release 1. Here are the major changes to this document from the previous edition, for Version 12 Release 1, SC19-3674-00.

- XML templates. In particular, see the new section “XML templates” on page 194, and the new appendix Appendix B, “XML definitions for a template,” on page 1197.
- New panels “Field Selection List panel” on page 570, “FMAP Copybook or Template panel” on page 578, “Template Export Utility panel” on page 689, and “Template Import Utility panel” on page 692.
- New primary commands “CCSID primary command” on page 745 and “FMAP primary command” on page 779.
- New functions “TPEXP (Export template to XML)” on page 1064 and “TPIMP (Import template from XML)” on page 1068.
- New external REXX functions “SEGLEN” on page 1143 and “SEGOFF” on page 1144.
- Two new fields in the “Copy From panel” on page 491
- Able to append “K” to some prefix commands (see “Editor panel” on page 534).
- New messages.

Changes are indicated by a “|” changebar in the left margin of the page.

The source files for this document were migrated from one source type to another. This has resulted in minor formatting adjustments and other changes. These changes, and other small changes such as minor editorial clarifications, are not listed. Many are tagged with the changebar.

Part 1. Working with File Manager

Chapter 1. File Manager overview

IBM File Manager for z/OS (base component) provides comprehensive, user-friendly tools for working with Websphere MQ data, HFS files and QSAM, VSAM and IAM data sets. These tools include the familiar view, edit, copy and print utilities found in ISPF, enhanced to meet the needs of application developers.

This chapter provides an overview of the types of actions you can perform with File Manager, a list of the types of data sets upon which you can perform these actions and some concept information about templates, one of the main tools used to view and edit data sets in File Manager.

File Manager actions

You can use File Manager to display, edit, update, create, copy, compare, print and erase data. You can manipulate this data as logical units, accessed by the data set name; or as physical units, accessed at each physical disk or tape location. Adding to this capability is an interactive interface that allows you to view data in both TABLE and SINGLE RECORD formats.

When working with data sets, you can:

- View, change, sort, copy and print data as unformatted or formatted with copybook or dynamic templates
- Perform sophisticated data set comparisons
- Find and change data strings in a single data set or in multiple data sets
- Create new HFS files or QSAM, VSAM or IAM data sets and data set records
- Create REXX procedures for advanced record processing
- Use selected DFSORT control statements to control the Data Set Copy and Data Set Print functions
- Create batch jobs, REXX procedures or CLISTs to automate tasks

For many of these tasks, you can use File Manager *templates*. A template provides a logical view of a data set, based upon field definitions in a copybook, or else created dynamically. If you use a template, you can:

- Format data according to record structures defined in the copybooks, or dynamically define your own record structure
- Select the fields and records that you want to work with, adjust how fields are displayed, select fields for copying, and set the contents of fields for created data
- Code record identification and selection criteria within File Manager panels, or use File Manager's REXX interface to code complex criteria
- Copy data between fields of different data type and length; and
- Work with files containing multiple record structures

File Manager also provides utilities that allow you to manipulate data in other ways:

- With the Tape Utility you can:
  - Copy tape files, optionally changing the block size and record format
  - Copy a logical volume from an Exported Stacked Volume to a physical tape volume

  - Update tape records on-screen, while copying the records from one tape to another
  - Compare two tapes byte-by-byte
  - Bypass tape blocks with data-check errors
  - Summarize the contents of a tape
- With the Disk Utility, you can look at and edit a file, based on the contents of the data at each physical disk location. There is support for character and hexadecimal format. Because the access is done at the level of the physical track, all file types are supported, including BDAM. With the disk utilities, you can:
  - List data sets on a disk
  - Edit data on a specific track
  - Identify the disk extents of a data set
  - Search for data within a disk extent
- You can use the Object Access Method (OAM) Utility to list, view, print, update, erase, copy, back up and restore OAM objects.
- The IDCAMS front end (Catalog Services) allows you to control the OS/390® IDCAMS utility from user-friendly File Manager panels.

Data sets and records

File Manager provides production and development logical file manipulation for HFS files, sequential (PS and PDS(E)) and VSAM (including IAM) data sets.

You can view and edit or otherwise process these types of data:

- Sequential (PS) and partitioned (PDS or PDSE) data sets: A member of a partitioned data set or a sequential data set, or like or unlike concatenations thereof (when used as input), with any of these data set attributes:
  - PS data set type
    - Basic
    - Library
    - Extended
    - Large
    - Compressed
  - Record format
    - Fixed, variable, or undefined
    - Blocked or unblocked
    - Spanned
    - ISPF packed format data
  - Record length
    - For fixed-length data: 1–32760, 10-65535 (for tape using LBI)
    - For variable-length data: 1–32752, 14-65531 (for tape using LBI)
    - For spanned variable-length data: 1–16MB
- VSAM data sets:
  - ESDS, including spanned format
  - KSDS, including spanned format
  - RRDS (fixed-length or variable-length)
  - PATH (PATHs related to a non-unique alternate index are restricted to browse only)
  - AIX®
- HFS files: A file contained in the z/OS UNIX File System, including z/FS and HFS:
  - Record format

    - Text (delimited by CR,LF or NL)
    - Binary
  – Record length
    - Text mode files are variable length: 1-32752
    - Binary mode files are fixed length: 1-32760
- Websphere MQ messages on a queue. Where allowed, an MQ queue can be specified in a data set name.
- CICS files, TS, or TD queues:
  – When used in conjunction with FM/CICS. See *File Manager User's Guide and Reference for CICS*.
  – In a non-CICS File Manager environment, by specifying the CICS resource type, CICS applid, and CICS resource name as the data set name.

Note: Throughout this publication, where the term *data set* is used, it generally refers to any of the above resource types, rather than referring specifically to traditional z/OS data sets, unless otherwise mentioned.

File Manager Version 13 is able to use copybooks that are stored in a PDS, a PDSE, a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Templates

A File Manager template is a file that provides a particular interpretation of a data set. The template defines a logical view that you can use to view and manipulate the data in records as discrete fields. With a template, you can select and format records and fields when editing, viewing, copying, printing and creating data sets. Templates can be based upon a COBOL, HLASM, OR PL/I*copybook* or can be created *dynamically*. You can apply different templates to the same data, to achieve different purposes. For example, you might create a number of templates with different selection criteria in place, and then apply the template that defines the subset of records that you need for a particular purpose. Also, you can apply the same template to different data sets, provided that they have similar record and field structures.

The File Manager Base, DB2 and IMS components all use templates. However, these templates are not interchangeable between the components. You cannot, for example, use a "base" template in the File Manager DB2 component. Dynamic templates are only available within the Base component.

Why use a template?

Typically, data sets containing application data have a well-defined structure, where each record in the data set consists of several fields. Each field contains a discrete item of data, such as a person's name, a product code or a currency amount (depending on the application). This data can be of various types: some fields contain character data, others contain numeric data (perhaps encoded in binary format to conserve storage space).

To work efficiently with these data sets, you need tools that recognize record structure, allowing you to selectively process individual fields.

For example, if you use a text editor (such as the ISPF editor) to display an application data set, then each record appears as a string of characters, with no indication that the records consist of individual fields. Fields containing numeric

data encoded in binary or packed decimal format are represented by their (typically "non-displayable") character values, rather than their numeric values.

Even if you know the column boundaries of each field, editing an application data set in a text editor can be difficult and tedious:

- If you want to find and replace data in a specific field, then you have to be careful not to change the field length; otherwise, you could move the boundaries of subsequent fields, and corrupt the record structure.
- If you want to find and replace data in numeric fields where the data has been encoded in binary or packed decimal format, then you have to specify the find and replace values in that format (typically, as hexadecimal values), rather than as numeric values.
- If the file you are editing contains several record types (each with its own structure), and you only want to find and replace data in one record type, then you have to exclude other record types from editing.

To illustrate, here are some records containing a mixture of alphanumeric and numeric fields displayed in the ISPF editor:

```
EDIT       USERID.FMDATA(DATA2) - 00.02                Columns 00001 00072
000013 01Grant Smith              Ī
000014 01Andrew Apple         5   Ī&
000015 01Graham Prestcott     õ   ç
```

The Grant Smith record contains the numeric value "94", stored in a binary numeric field. To change this from 94 to 48, without using a template, you would need to:

1. Display the hexadecimal values of the record data.
2. Determine the column positions occupied by the field.
3. Convert the numeric value 48 into its hexadecimal equivalent (X'30').
4. Overtype the old hexadecimal value of the numeric field (X'5E') with the new value, or use a CHANGE command with the old and new hexadecimal values.

Here is the same record displayed in the File Manager editor using a template:

```
REC-TYPE NAME                EMPLOYEE-NO     AGE   SALARY  MONTH(1)  MONTH(2)
#2       #3                         #4      #5       #6        #7        #7
AN 1:2   AN 3:20               BI 23:2 BI 25:2  PD 27:4   BI 31:4   BI 35:4
<>       <---+----1----+---->     <---+-> <---+-> <---+--> <---+---> <---+--->
01       Grant Smith                7712      94    75000         6        15
```

To change the value of the AGE binary numeric field from 94 to 48, you can simply overtype 94 with 48, or enter the following command (where #5 is the "field reference" for the AGE field):

```
CHANGE 94 48 #5
```

Another situation in which you might want to use a template is when you are copying data. When you use a copy utility that does not recognize record structure, you are often limited to copying an entire data set, or a number of records. If you want to copy only records with particular field values, or only particular fields, then you must write a custom program.

When using templates with File Manager, you can specify criteria to select only the records you want to copy, and to select the fields you want from the list of fields in the record.

What can you do with a template?

Using a template, you can:

- **Format records**

  You can display, print and compare fields taking into account their data types. For example, binary numeric fields are displayed as their true numeric values, not as the character representation of their binary value.

  When editing, you can overtype or use CHANGE commands to replace these formatted numeric field values with different numeric values; File Manager adjusts the underlying binary value in the data set.

  You can select which fields are displayed (when in an editor session) or printed. In an editor session, you can limit the scope of commands (such as FIND and CHANGE) to particular fields.

  You can change the order in which fields are displayed (when in an editor session) or printed.

  For numeric fields, you can display or suppress leading zeros (so that, for example, 00057 is displayed and printed as 57); the default is to suppress leading zeros.

- **Reformat records**

  When copying data, you can "map" which fields in the input data set you want copied to the output data set. You can insert or delete fields, and copy data between fields of different data types or lengths.

- **Identify record types and select records within those types**

  You can select records using two levels of criteria:

  – Record identification criteria

    Identify the unique characteristics of a record type, enabling File Manager to distinguish it from other types of record in the same data set. You can then select which record types in a data set you want to use.

    Before you can specify record identification criteria, you need to examine your data, and determine what field values uniquely identify a record as belonging to a particular type.

  – Record selection criteria

    After a record has been identified as belonging to a particular type, you can specify record selection criteria to narrow down which records of that type you want to use.

- **Create data**

  When creating a new data set, or adding a new field when copying a data set, you can specify field "create attributes" (patterns that File Manager uses to initialize field values).

Where can you use templates?

The following table lists the tasks, panels and functions that support the use of templates:

Task	Panel	Function
View data	View (option 1)	—
Edit data	Edit (option 2)	DSEB (batch only)[1]
Compare data	Compare data (option 3.11)	DSM
Copy data	Copy Utility (option 3.3)	DSC

Task	Panel	Function
Create data	Data Create Utility (option 3.1)	DSG
Print data	Print Utility (option 3.2)	DSP
Update data	—	DSU (batch only)[1]

Note:

1. For the DSEB and DSU functions, the template is not used during editing or updating; it is only used during enhanced batch processing, when you call the File Manager REXX external function PRINT with SNGL or TABL format.

   The difference between using DSU to *update* data compared to using DSEB to *edit* data is:

   - DSU operates on a "per record" basis. That is, each record in a data set is treated separately, in sequence.
   - DSEB allows you to move between the records in a data set as you choose, similar to the way you can scroll around a data set in an edit panel.

   For details on using the DSEB and DSU functions, see Chapter 13, "Enhancing File Manager processing," on page 405.

With any of these panels or functions, you can supply an existing template (created dynamically or from a copybook), supply a copybook, or create a new dynamic template. If you supply a copybook, then File Manager compiles it into a template before use.

Chapter 2. Getting started with File Manager

This chapter is designed to help the new user to familiarize themselves with the user interface and general methods for working within File Manager panels.

The major tasks described in this chapter are:

Starting and exiting File Manager

File Manager is an ISPF-developed application and is usually run from within an ISPF session. Exiting from File Manager returns you to your ISPF session. File Manager panels can also be invoked from outside of the File Manager application.

A subset of File Manager functions is available in non-ISPF mode (batch processing). See Chapter 16, "Functions," on page 831.

The tasks described in this section are:

Starting File Manager

For information about running File Manager in non-ISPF mode (batch processing), see Chapter 16, "Functions," on page 831.

How you start File Manager in ISPF depends on how it has been installed on your system. Typically, File Manager is installed as an option on your z/OS Primary Option Menu:

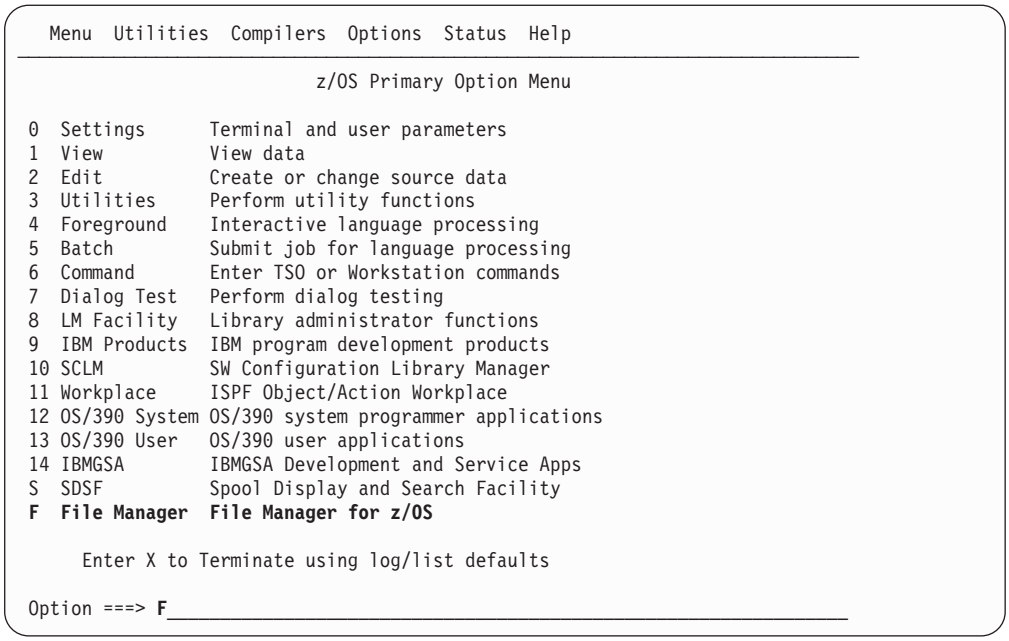


Figure 2. z/OS Primary Option Menu panel showing File Manager option

To start File Manager:

1. Select the File Manager option from the z/OS Primary Option Menu.

   Note: If File Manager is not an option on your menu, ask your system administrator to explain the startup process used at your site.

   When you start File Manager, the File Manager Primary Option Menu is shown, with the current version information displayed in a pop-up message box.

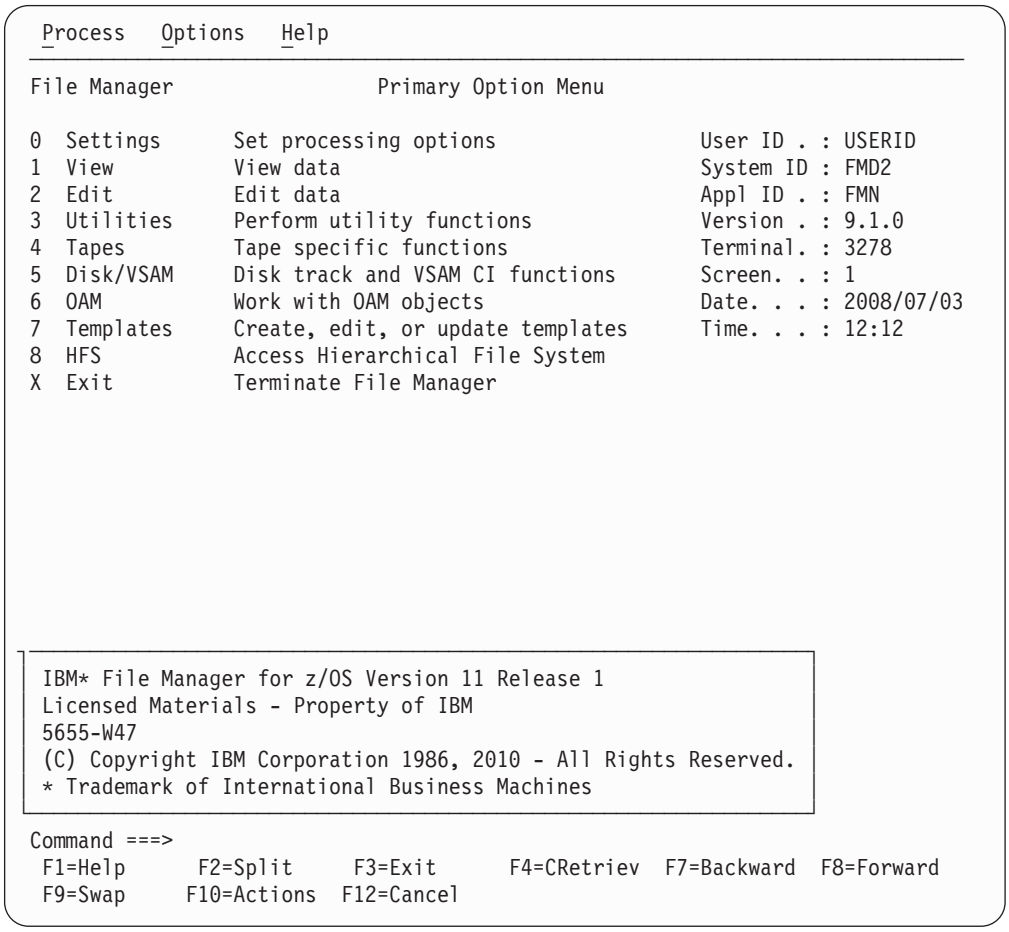


Figure 3. File Manager Primary Option Menu panel with version information

2. Press Enter to close the version window.

Related topics

"Primary Option Menu panel" on page 615

Checking your File Manager version

To display the full version information for File Manager, enter VER on the Command line of any panel. The current File Manager version number and the PTF number for each File Manager component is displayed in a window.

As File Manager is an ISPF application and ISPF applications are typically not capable of running in an authorized state, File Manager will not (unless specific ISPF customization has occurred) be authorized under ISPF. However, File Manager may run in an authorized state when used in a batch job.

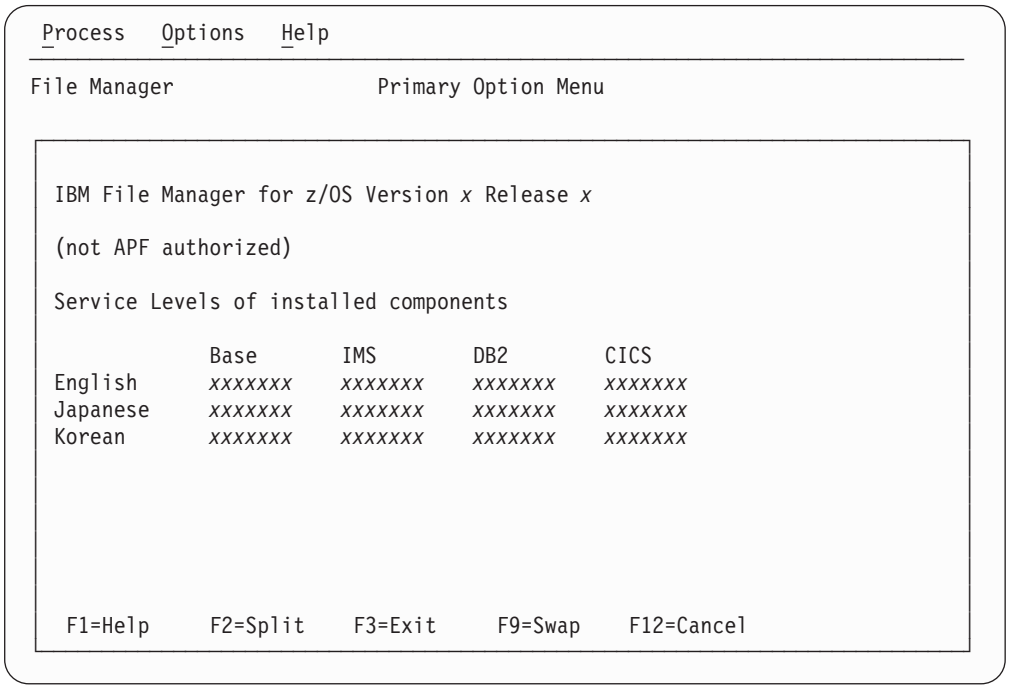
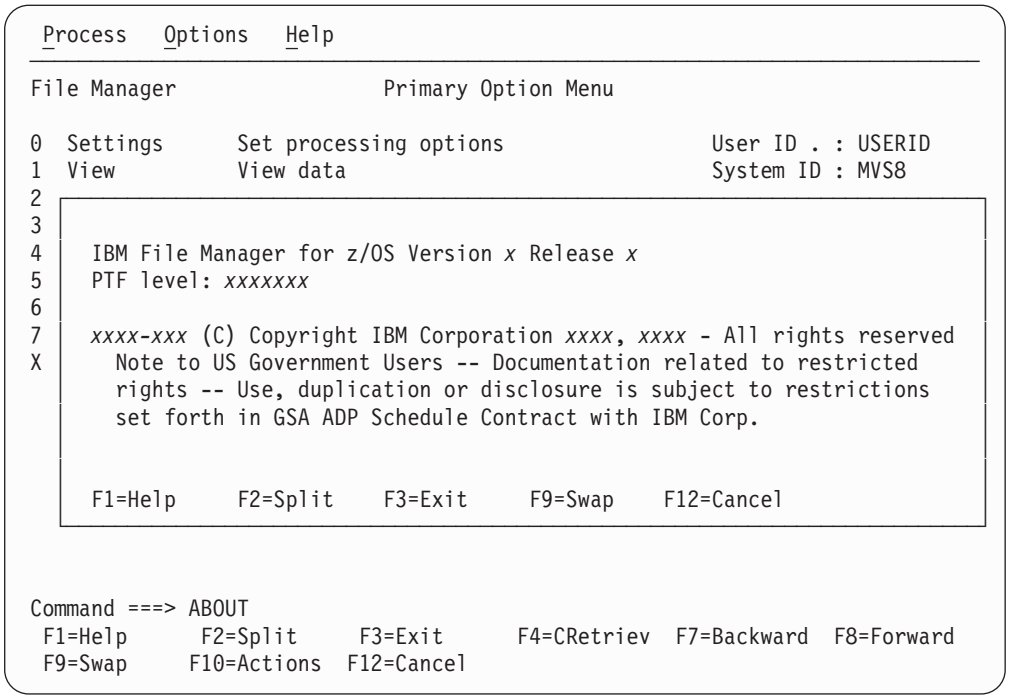


Figure 4. File Manager Primary Option Menu panel with example PTF information

An alternative way to display version information is to select **Help** > **6. About** from the Action Bar on any panel. The current File Manager version number and the PTF number of the File Manager base component is displayed. The window also shows the copyright information and any notes from IBM that are shipped with the product.



Related topics

"VER primary command" on page 828
"ABOUT primary command" on page 739
"VER (Display Service Level)" on page 1085

Checking which COBOL compiler you're using

File Manager uses the currently loaded COBOL compiler to create a template from a COBOL copybook.

To display details about the current COBOL compiler, enter SHOWCOB on the Command line of any panel. File Manager displays details about the current COBOL compiler in a window.

Related topics
"SHOWCOB primary command" on page 817

Exiting from File Manager

You can exit from File Manager from the Primary Option Menu panel in any of the following ways:

- Press the Exit function key (F3).
- Enter `X` (or `EXIT` or `END`) on the Command line.
- Select **Process**> **Exit from File Manager** from the Action Bar.

To exit the application from any panel within File Manager:

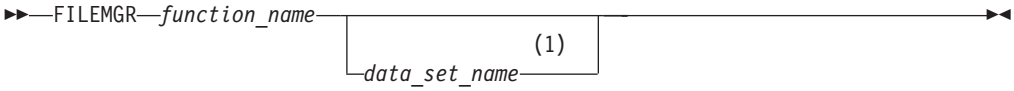
- Enter `=X` on the Command line.

Related topics
"END primary command" on page 762
"EXIT primary command" on page 766

Invoking File Manager panels from outside File Manager

File Manager panels can also be invoked directly from any REXX procedure, TSO clist or ISPF Command line. To do this, enter the function name for the required File Manager panel (as listed in the "Equivalent function" column in "Summary of File Manager panels" on page 433). This method of invoking File Manager causes ISPF to display a panel for the specific function allowing you to enter the values required for function processing.

Syntax



Notes:

1 The *data_set_name* parameter is only allowed for the DSV function, to invoke View (option 1), and the DSE function, to invoke Edit (option 2).

For example, from any ISPF Command line, you can enter the following command to display the File Manager Print Utility (option 3.2) panel:

```
FILEMGR DSP
```

If you specify a valid data set name after DSV (Data Set View) or DSE (Data Set Edit), File Manager:

- Bypasses the entry panel (where you would normally specify the data set name and other options, such as the name of a stored template you might wish to use when displaying the data set).

- Goes directly to viewing or editing the specified data set.

Related topics
"Using File Manager functions in batch jobs" on page 401
"Summary of File Manager panels" on page 433
Chapter 16, "Functions," on page 831
"DSB (Data Set Browse)" on page 868
"DSE (Data Set Edit)" on page 900

Using the File Manager interface

The File Manager interface is based upon the ISPF model. Tasks are performed by processing a function, together with its applicable parameters. Panels provide a user-friendly way of selecting a function and supplying the parameter information. Most functions can be processed in your choice of "foreground" or "batch" mode. In foreground mode, the selected function is processed immediately and any results are returned to you on screen or directed to print outputs, as dictated by your default settings. In batch mode, JCL is generated from the panel information, and then presented to you for editing. You can modify this code and submit it to a processing queue.

The tasks described in this section are:
"Navigating in File Manager";
"Issuing commands" on page 15;
"Specifying a data set and a member name" on page 16; and
"Using hexadecimal values in File Manager" on page 37.

Navigating in File Manager

You navigate around File Manager panels in the same way as any other panels under ISPF. That is, you launch processing panels from menu panels, by entering the appropriate menu option number on the Command line. File Manager uses a Primary Options Menu panel to provide access to processing panels that are related to a particular function or group of functions. In some cases, an option on the Primary Options menu leads to another menu panel, from which you can select the required processing panel.

When you know the menu structure well, you can quickly jump to the required panel by entering the full pathway through the menu options. For example, entering `4.2.8` would take you directly to the Exported Stacked Volume Copy panel.

You can also jump to any File Manager panel from any other panel, without having to navigate the menu structure, by using an equals sign (=) to prefix the option number. The equals sign instructs File Manager to start the menu navigation from the Primary Options menu and not from your current location. For example, if you were positioned on the Set Processing Options Menu panel, entering 2 on the Command line would display the Set System Processing Options panel (option 2 on the current menu), while entering `=2` would display the Edit Entry panel (option 2 on the Primary Options Menu panel).

To access a panel, choose one of the following methods:

- Type the menu option number on the Command line and press Enter. Repeat for each nested menu until the required processing panel is displayed.
- Type an equals sign followed by the complete menu path to the required panel, separating each menu level with a period, then press Enter.

Many File Manager panels have more information than can be seen in a single screen, particularly when you are running in 80x24 display. When this happens, a message is displayed in the top right corner of the panel.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                Data Create Utility
                                                                  More:     +
 Output Partitioned, Sequential or VSAM Data Set, or HFS file:
```

Figure 5. Example panel showing "More" message

To scroll down the panel, choose one of these methods:

- Press the Forward function key (F8) (this scrolls down by your default scroll amount, usually one screen at a time)
- Enter FORWARD or DOWN on the Command line (these commands scroll down by your default scroll amount, usually one screen at a time)
- Enter BOTTOM on the Command line (this takes you to the bottom of the panel)

To scroll back to the top of the panel, choose one of these methods:

- Press the Backward function key (F7) (this scrolls up by your default scroll amount, usually one screen at a time)
- Enter BACKWARD or UP on the Command line (these commands scroll up by your default scroll amount, usually one screen at a time)
- Enter TOP on the Command line (this takes you to the beginning of the panel)

To exit from any panel, choose one of the following:

- Press the Exit function key (F3).
- Type `X` on the Command line and press Enter.
- Type `END` on the on the Command line and press Enter.

Related topics

“Primary Option Menu panel” on page 615
“Scrolling to see data” on page 77
“UP primary command” on page 826
“DOWN primary command” on page 760
“TOP primary command” on page 824
“BOTTOM primary command” on page 742

Issuing commands

Within the File Manager panels, you can use "primary" or "prefix" commands to instruct File Manager to perform an action.

You enter primary commands on the Command line of File Manager panels. On many of these panels, the most commonly used primary commands have been assigned to function keys, also known as Programmable Function keys or PF keys.

Some primary commands are exclusive to a particular panel, although most are available on several panels. Where a command is available on more than one panel, its behavior might differ depending upon the context in which it is used.

You enter prefix commands in a special prefix field in most panels, and in the prefix area within the editor panel. Prefix commands are generally exclusive to a panel and details of their operation are provided within each panel description.

Related topics
Chapter 15, “Primary commands,” on page 739
Chapter 14, “Panels and fields,” on page 433

Specifying quoted strings

Throughout the File Manager interface, you might encounter places where you need to specify a *quoted string*, for example, fully qualified data set names, criteria expressions and so on. In these situations, you can use either the double quotation symbol ("), also called simply a quotation mark, or the single quotation symbol ('), also called an apostrophe. However, you must use matching symbols to open and close the quoted string. For example:

```
"fmndata.test1" and 'fmndata.test1'
```

are both legitimate ways to specify a data set.

The simplest way to include an apostrophe or quotation symbol as a character in your string, is to use the opposite character as the string delimiters. For example:

```
"it's"
OR
'he said, "hello"'
```

When your string contains a mix of apostrophes and quotation marks, use two consecutive quotation marks ("") to represent a " character within a string delimited by quotation marks, or two consecutive apostrophes ('') to represent a ' character within a string delimited by single quotation marks. For example, to find the string, "he said, "Take it it's yours.", you would type:

```
FIND 'he said, "Take it it''s yours."'
```

Note: This document uses the words *quotation marks*, or *quotes* to mean either " or '.

Specifying a data set and a member name

Many File Manager panels require you to specify the data set that you wish to use for the current process. When the data set is a PDS or a CA-Panvalet or other external library, you must also specify the member name. These panels are collectively referred to as Entry panels, for example, the View Entry panel.

All Entry panels contain the following fields:

```
⋮
Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name . . . . . _____________________________________________+
   Member  . . . . . . . . . .  ________      Blank or pattern for member list
   Volume serial . . . . . . .  ______        If not cataloged
⋮
```

Figure 6. Data set/path name specification fields

To specify your data set on any Entry panel:

1. In the Data set/path name field, enter a path name ,fully-qualified data set name, or a name pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated Member field must be empty.

For details about rules for specifying a path name, see "Specifying an HFS file" on page 391.

In the case of a copybook, this can be the name of a CA-Panvalet library or external library accessed using the Library Management System Exit.

If you do not enclose the data set/path name in quotes, then the default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

If you enter a pattern, when the Entry panel is processed, File Manager displays a list of the data sets that match the pattern.

Patterns can contain a percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.

For performance reasons, you should qualify the data set/path name as much as you can.

2. If you specified the name of a partitioned data set (PDS) or a CA-Panvalet or other external library in the **Data set/path name** field (without including a member name or name pattern in parenthesis), then perform either of these actions:
   - Enter a name or name pattern in the **Member** field.

     A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

     asterisk (*)
     Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter `*d*`, a list of all members in the data set whose name contains "d" is displayed.

     percent sign (%)
     A place-holding character representing a single character. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter `%%%`, a list of all members in the data set/path whose name is four characters in length is displayed.

     If you perform either of these actions
     - Do not include a member name or name pattern in parenthesis in the **Data set/path name** field, and leave the **Member** field blank or you specify a pattern in the **Member** field,
     - Include a member name pattern in parenthesis in the **Data set/path name** field and leave the **Member** field blank,

     File Manager displays a member name list when the Entry panel is processed and the data set has been specified or selected.
   - If the entry panel you are using includes the option, **Advanced member selection**, and you want to select a range of PDS(E) members based on one or more criteria,select that option to display the Advanced Member Selection panel.

     For details about selecting a range of PDS(E) members, see "Selecting a range of PDS(E) members" on page 34.
3. For data sets which have not been cataloged, enter the serial number of the volume which contains the data set in the **Volume serial** field.
4. Complete the remaining panel-specific fields, and then press Enter to process the panel function.

If both the data set/path name and the member name have been fully specified, the panel action proceeds to its next stage.

If you entered a pattern for the data set/path name or member name, the Data Set Selection panel is displayed, followed by the Member Selection panel, as needed. When you have made your selections from these panels, the Entry panel action proceeds to its next stage.

5. When the Data Set Selection panel is displayed, select your data set by entering an `S` in the **Sel** field adjacent to the required data set. You can only select one name from this list.
6. When the Member Selection panel is displayed, select your members by typing an `S` in the **Sel** field next to each member or with the SELECT primary command, and then pressing Enter.

   You can select as many names as required from this list. When the panel is processed, the first member on the list is used in the relevant function. When the function is completed, the next member on the list is used. For example, if you are selecting several members to be edited, the first member is displayed in the Edit Session panel. When you exit this panel, the next member is displayed, and so on, until all selected members have been processed.

   Note:

   a. To reset values entered in the prefix area that have not yet been processed, use the RESET primary command.
   b. To re-read the directory and display the current member list, use the REFRESH primary command.
7. When you are selecting members, you might find it useful to sort the list of displayed members. To do this, perform either of these actions
   - Use the SORT primary command to sort the member list by up to two of the displayed columns. The field names are the column headings. For example, `SORT SIZE CREATED` sorts the list of members first by size and then by the date created.
   - Place the cursor on the column header of the column you want to sort and press Enter.

   Note: When you sort members, the sequence (ascending or descending) for a given column is predetermined and consistent with ISPF.

Examples of data set name patterns

The examples below assume that the following data sets exist under the USERID high-level qualifier:

```
USERID.FMN.DATA
USERID.FMN.TEMPLATES
USERID.COBOL.COPY
USERID.COBOL.SOURCE
USERID.PLI.COPY
USERID.PLI.SOURCE
USERID.MISC.DATA.BACKUP
USERID.WORK
```

Example 1

Entering the following pattern (with a single asterisk):

```
'USERID.*'
```

displays the only item with exactly one level of qualifier after USERID:

```
USERID.WORK
```

Example 2

Entering the following pattern (with two asterisks):

```
'USERID.**'
```

displays the complete list of data sets with the USERID high-level qualifier.

Example 3

Entering the following pattern:

```
'USERID.*.%%%%'
```

displays the following list (containing those data sets with exactly three qualifiers, whose third-level qualifier contains exactly four characters):

```
USERID.FMN.DATA
USERID.COBOL.COPY
USERID.PLI.COPY
```

Examples of specifying a member name or pattern

The examples below assume that the following members exist within the data set name, FMNUSER.DATA:

```
DATA1
FMNCDATA
FMNCTAM
FMNCTEM
FMNCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Example 1

Entering the following details on an Entry panel:
displays the entire member name list:

```
:
   Data set name . . . . . FMNUSER.DATA
   Member  . . . . . .
:
```

```
DATA1
FMNCDATA
FMNCTAM
FMNCTEM
FMNCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Example 2

Entering the following details on an Entry panel:

```
:
   Data set name . . . . . FMNUSER.DATA
   Member  . . . . . . . . FMNCT*
:
```

displays the member name list:

```
FMNCTAM
FMNCTEM
FMNCTMP
```

Example 3

Entering the following details on an Entry panel:
displays details for the member NEWDATA in data set FMNUSER.DATA.

```
:
   Data set name . . . . . FMNUSER.DATA(NEWDATA)
   Member  . . . . . . . . ________
:
```

Example 4

Entering the following details on an Entry panel:
displays the member name list:

```
:
   Data set name . . . . . FMNUSER.DATA(TEM*)
   Member  . . . . . . . . ________
:
```

```
TEMPA
TEMPB
```

Example 5

Entering the following details on an Entry panel:
displays the entire member name list:

```
:
   Data set name . . . . . FMNUSER.DATA(*)
   Member  . . . . . . . . ________
:
```

```
DATA1
FMNCDATA
FMNCTAM
FMNCTEM
FMNCTMP
NEWDATA
NEWSTUFF
TEMPA
TEMPB
```

Related topics

"Selecting a range of PDS(E) members" on page 34
"Browse Entry panel" on page 459
"Edit Entry panel" on page 541
"Data Create Utility panel" on page 515
"Print Utility panel" on page 620
"Copy From panel" on page 491
"Copy To panel" on page 497
"Find/Change Utility panel" on page 574
"AFP Print Browse panel" on page 442
"Load Module Information panel" on page 592
"Compare Utility: "Old" and "New" panels" on page 471
"Template Workbench panel" on page 698
"Data Set Selection panel" on page 519
"Member Selection panel" on page 600
"SORT primary command" on page 818
"RESET primary command" on page 806
"REFRESH primary command" on page 804
"Selecting a range of PDS(E) members" on page 34

Specifying an MQ manager or queue

Many File Manager panels allow you to specify the name of an MQ manager or queue that you wish to use for the current process.

Where allowed, you specify the name of the MQ manager or queue in the data set name field in the form:

`MQ:managerid:queuename`

Where:

managerid
The ID of the MQ manager (the 4-character subsystem ID).

queuename
The name of a queue managed by that manager.

If you enter a pattern, when the Entry panel is processed, File Manager displays a selection list allowing you to select a manager or a queue.

Patterns can contain a percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.

Related topics

"Browse Entry panel" on page 459
"Compare Utility: "Old" and "New" panels" on page 471
"Copy From panel" on page 491
"Copy To panel" on page 497
"Create panel" on page 512
"Edit Entry panel" on page 541
"Find/Change Utility panel" on page 574
"Print Utility panel" on page 620
"View Entry panel" on page 713

Specifying a CICS resource

Many File Manager panels and batch functions allow you to specify the name of a CICS resource that you wish to use for the current process.

To specify a CICS resource in a non-CICS File Manager environment for base File Manager and Eclipse, use the notation shown here. The same notation applies to both panel and keyword fields. You can only specify generic values on panels.

```
rt:applid:rname
```

Where:

rt Resource type. Valid values are:
- **FI** For a CICS file.
- **TD** For a Transient Data Queue.
- **TS** For a Temporary Storage Queue.

applid[1] The VTAM® applid of the CICS system. You can specify a generic name to display a list of CICS systems if an FMNCICS DD describing CICS applids has been allocated to the ISPF session. See the *File Manager Customization Guide* for details.

rname The name of the resource. You can specify a generic name to select from a list.

Note:

1. To specify a CICS resource from File Manager when running under FM/CICS, omit the *applid* parameter from the resource name:

   ```
   rt:rname
   ```

Patterns (only usable on panels) can contain a percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.

For a list of functions that allow you to access CICS resources from ISPF using the File Manager editor or browser, see Table 1.

Related topics

"Accessing CICS resources"
"Browse Entry panel" on page 459
"Compare Utility: "Old" and "New" panels" on page 471
"Copy From panel" on page 491
"Copy To panel" on page 497
"Create panel" on page 512
"Edit Entry panel" on page 541
"Find/Change Utility panel" on page 574
"Print Utility panel" on page 620
"View Entry panel" on page 713

Accessing CICS resources

You can access CICS resources from File Manager base. Table 1 shows the functions that allow you to specify a CICS resource.

Table 1. Functions allowing access to CICS resources

Option	Description	Function
	Browse	DSB
1	View[1]	DSV
2	Edit[1]	DSE
3.1	Create	DSG

Table 1. Functions allowing access to CICS resources (continued)

Option	Description	Function
3.2	Print	DSP
3.3	Copy	DSC
3.6	Find/Change	FCH
3.11	Compare	DSM
10	FM/CICS functions[2]	
	Data Set Edit Batch	DSEB
	Data Set Update	DSU

Note:

1. Plus commands: APPEND, COPY, CREATE, REPLACE, SAVEAS.
2. See *File Manager User's Guide and Reference for CICS*.

Examples of accessing CICS resources

Shown here are two examples of accessing CICS resources.

Example 1: Searching and editing a CICS Temporary Storage Queue

To search and edit a CICS Temporary Storage Queue, follow these steps:

1. From the Primary Option Menu panel, select option 2 (Edit) to display the Edit Entry panel.
2. In the **Data set/path name** field, type "TS:" as shown in Figure 7.

   The "TS:" indicates to File Manager that you want to edit one or more *Temporary Storage Queues* from a CICS application.

   Note: To edit CICS *Transient Data Queues*, type "TD:"; to edit CICS *files*, type "FI:".)

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Edit Entry Panel

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name TS:                                          +
   Member . . . . . ________     Blank or pattern for member list
   Volume serial  . ______       If not cataloged
   Start position . ____________________________ +
   Record limit . . ________     Record Sampling _
   Inplace edit . . _              (Prevent inserts and deletes)
Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . . . . __________   Blank or pattern for member list

Processing Options:
 Copybook/template   Start position type   Enter "/" to select option
 3  1. Above         _  1. Key             _  Edit template   _ Type (1,2,S)
    2. Previous         2. RBA             _  Include only selected records
    3. None             3. Record number   _  Binary mode, reclen _____
    4. Create dynamic   4. Formatted key   _  Create audit trail
                                           _  Use I/O exit
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 7. Specifying CICS temporary storage queues

3. Press Enter.

When you specify the type of CICS resource ("TS:", "TD:", or "FI:") on an entry panel, but without any further qualification, File Manager next displays the CICS Applid Selection List panel which lists the available CICS applications as shown in Figure 8.

```
 Process   Options   Help
-------------------------------------------------------------------------------
 File Manager           CICS Applid Selection list          Row 00001 of 00023

  Applid   Status   Description
  *        *        *
_ CICSDEV1 Active   Development system 1
_ CICSDEV2 Active   Development system 2
_ CICSTST1 Inactive Testing system 1
_ CICSTST2 Inactive Testing system 2
_ CICSTST3 Inactive Testing system 3
_ CICSTST4 Active   Testing system 4
_ CICSPRD1 Inactive Production system 1
_
:

Command ===> ___________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 8. List of CICS applications

4. Select the CICS application you want by typing "S" in the entry field next to the relevant applid (in this case, applid CICSTST1) as shown in Figure 9.

```
 Process   Options   Help
-------------------------------------------------------------------------------
 File Manager           CICS Applid Selection list          Row 00001 of 00023

  Applid   Status   Description
  *        *        *
_ CICSDEV1 Active   Development system 1
_ CICSDEV2 Active   Development system 2
S CICSTST1 Inactive Testing system 1
_ CICSTST2 Inactive Testing system 2
_ CICSTST3 Inactive Testing system 3
_ CICSTST4 Inactive Testing system 4
_ CICSPRD1 Inactive Production system 1
_
:

Command ===> ___________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 9. Selecting a CICS application

5. Press Enter.

   File Manager then displays a list of Temporary Storage Queues (because you specified "TS:" on the Edit Entry Panel) for the selected CICS application as shown in Figure 10 on page 25.

```
 Process   Options   Help
 File Manager      CICS Temporary Storage Selection List    Row 00001 of 00005

  Queue            Loc   Items Size      Max   Min   Tran   Last     SYS   Po ±
  *                *                                  *               *     *
_ lowercase        AUX      20      6400   320   320 FM        97210
_ KSDSXX           AUX      20      2560   128   128 FM        98913
_ KSDS10           AUX      10      1280   128   128 FM        14654
_ TDTD"'TDTD TD    AUX     100     32000   320   320 FM        97194
_ TDTEST10         AUX      10      3200   320   320 FM        97423
****  End of data  ****

Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 10. List of CICS temporary storage queues

6. Select the Temporary Storage Queue you want by typing "S" in the entry field next to the relevant queue (in this case, KSDS10) as shown in Figure 11.

```
 Process   Options   Help
 File Manager      CICS Temporary Storage Selection List    Row 00001 of 00005

  Queue            Loc   Items Size      Max   Min   Tran   Last     SYS   Po ±
  *                *                                  *               *     *
_ lowercase        AUX      20      6400   320   320 FM        97210
_ KSDSXX           AUX      20      2560   128   128 FM        98913
S KSDS10           AUX      10      1280   128   128 FM        14654
_ TDTD"'TDTD TD    AUX     100     32000   320   320 FM        97194
_ TDTEST10         AUX      10      3200   320   320 FM        97423
****  End of data  ****

Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 11. Selecting a CICS temporary storage queue

7. Press Enter.

   File Manager then displays the data in the selected Temporary Storage Queue as shown in Figure 12 on page 26.

```
 Process   Options   Help
------------------------------------------------------------------------------
Edit             TS:CICSTST1:KSDS10                              Top of 10
       Col 1       Insert Length 80              Record AT TOP     Format CHAR
       ----+----10---+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 ....1       aaaaaaaaaaaaaa                          6324West aindy    Wa
000002 ....2       aaaaaaaaaaaaaa                          6324West aindy    Wa
000003 ....3       74 Redcliffe St                         6324West aindy    Wa
000004 ....4       74 Redcliffe St                         6324West aindy    Wa
000005 ....5       74 Redcliffe St                         6324West aindy    Wa
000006 ....6       74 Redcliffe St                         6324West aindy    Wa
000007 ....7       74 Redcliffe St                         6324West aindy    Wa
000008 ....8       74 Redcliffe St                         6324West aindy    Wa
000009 ....9       74 Redcliffe St                         6324West aindy    Wa
000010 ....10      74 Redcliffe St                         6324West aindy    Wa
****** ****  End of data  ****

⋮
```

Figure 12. Data in CICS temporary storage queue

Example 2: Searching and modifying a CICS file

To search and modify a CICS file, follow these steps:

1. From the Primary Option Menu panel, select option 2 (Edit) to display the Edit Entry panel.
2. In the **Data set/path name** field, type "FI:CICSTST1" as shown in Figure 13.

   The "FI:" indicates to File Manager that you want to edit one or more CICS *files*, and the "CICSTST1" indicates that you want to edit the files from the CICS application, CICSTST1.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Edit Entry Panel

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name FI:CICSTST1                                 +
   Member . . . . . ________     Blank or pattern for member list
   Volume serial  . ______       If not cataloged
   Start position . ____________________________ +
   Record limit . . ________     Record Sampling _
   Inplace edit . . _              (Prevent inserts and deletes)
Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . . . . __________   Blank or pattern for member list

Processing Options:
 Copybook/template   Start position type   Enter "/" to select option
 3  1. Above          _  1. Key            _  Edit template   _ Type (1,2,S)
    2. Previous          2. RBA            _  Include only selected records
    3. None              3. Record number  _  Binary mode, reclen _____
    4. Create dynamic    4. Formatted key  _  Create audit trail
                                           _  Use I/O exit
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 13. Specifying CICS files from a CICS application

3. Press Enter.

   When you specify the type of CICS resource ("TD:", "TS:", or "FI:") *and* the CICS application on an entry panel, but without specifying the name of the CICS resource, File Manager next displays a selection list of the Transient Data Queues, Temporary Storage Queues, or files for the specified CICS application.

In this case, File Manager displays a list of CICS files for the specified CICS application (CICSTST1) as shown in Figure 14.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager           CICS File Selection List          Row 00001 of 00020
  File     Data Set Name                                 Type O E R U A B D SY ±
  *        *                                             *    P N E P D R E *
                                                              E A A D D O L

 _ DFHCSD   CICS.FILE.DFHCSD                            KSDS O E R U A B D
 _ DFHDBFK                                              VSAM C E R U A B D
 _ DFHLRQ   CICS.FILE.DFHLRQ                            KSDS O E R U A B D
 _ ESDS10   CICS.FILE.ESDS                              KSDS O E R U _ B _
 _ EZACACHE EZACACHE                                    TABL C U R U A B D
 _ EZACONFG CICS.FILE.EZACONFG                          KSDS C D R _ _ B _
 _ FMDATA   CICS.FILE.FMDATA                            KSDS C E R _ _ _ _
 _ PMRRDS1  CICS.FILE.PMRRDS1                           KSDS C E R _ _ _ _
 _ RRDS10   CICS.FILE.RRDS10                            KSDS O E R U _ B _
 _
⋮

 ****  End of data  ****

 Command ===> _____________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
  F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 14. Selection list of CICS files

4. You can now proceed to work with any of the CICS resources listed.

   For more information about working with CICS resources in File Manager, see *File Manager User's Guide and Reference for CICS*.

Running File Manager batch functions with a CICS resource

When you run a File Manager batch function with a CICS resource, you must specify the resource type, VTAM applid of the CICS system, and the name of the CICS resource. You cannot specify generic values. See "Specifying a CICS resource" on page 21.

Table 1 on page 22 shows you the File Manager batch functions you can use with CICS resources.

When you run File Manager batch functions with a CICS resource, be aware that:

- If you are specifying a batch keyword for a CICS TS queue that contains special or lowercase characters, or a CICS TD queue that contains lowercase characters, you must enclose the name of the resource in quotes like this:

  `c'rt:applid:rname'`

  Any single quote in the resource name must be repeated. For example, to specify the TS queue name 'quoted name', enter:

  `DSNIN=c'TS:MYCICS:''quoted name'''`

- For DSG and DSC functions:
  - The DISP=MOD parameter *appends to* the TD (transient data) and TS (temporary storage) queues. Records are always appended for CICS files.
  - The DISP=OLD parameter *replaces* the TD (transient data) and TS (temporary storage) queues. This is ignored for CICS files where the data is always appended

Shown here are some examples of running File Manager batch functions with a CICS resource.

Example 1: Data set Generate: Creating a TS queue using a template

```
$$FILEM DSG DSNOUT=TS:PRODCICS:MYTSQ
$$FILEM RECSIZE=2000,
$$FILEM DISP=OLD,
$$FILEM TCOUT=MY.TEMPLATE.DSN(MYTP),
$$FILEM NLRECS=100
```

Example 2: Data Set Copy: Copying a flat file to a CICS intrapartition TD queue

```
$$FILEM DSC DSNIN=MYHLQ.QUEUE.DATA,
$$FILEM DSNOUT=TD:PRODCICS:MYIQ
```

Example 3: Data Set Print: Printing a CICS file using a template

```
$$FILEM DSP FORMAT=TABL,
$$FILEM TCIN=MY.TEMPLATE.DSN(MYTP),
$$FILEM DSNIN=FI:PRODCICS:MYKSDS
```

Example 4: Find/Change: Making changes to a CICS file

```
$$FILEM FCH ,
$$FILEM DSNIN=FI:PRODCICS:MYKSDS
CAPS ON
C 'ABC' 'DEF' 12 14
/+
```

Example 5: Data Set Edit Batch: Navigating a CICS file backwards and changing a name

```
$$FILEM DSEB DSNIN=FI:PRODCICS:MYKSDS,PROC=*
BOT()
DO UNTIL(UP(1) = 'TOF')
  IF FLDI(3,'=','JOHN') THEN DO
    OVLY_OUT('BILL',3)
    UPDATE()
    LEAVE
  END
END
/+
```

Example 6: Data Set Copy: Copying a flat file into a temporary storage queue

```
$$FILEM DSC DSNIN=myhlq.flatfile,
$$FILEM DSNOUT=TS:TESTCICS:MYTSQ
```

Example 7: Data Set Compare: Comparing two files on different CICS systems

```
$$FILEM DSCMP TYPE=RECORD,
$$FILEM SYNCH=ONETOONE,
$$FILEM LIST=DELTA,
$$FILEM WIDE=YES,HILIGHT=YES,
$$FILEM DSNOLD=FI:PRODCICS:MYKSDS,
$$FILEM DSNNEW=FI:TESTCICS:MYKSDS
```

Manipulating your view of selection lists

When you are working with selection lists, you can manipulate your view of the information available to focus on the items of interest to you.

To return to the original view of the selection list, enter the RESET primary command.

To rebuild the information required to display the selection list, enter the REFRESH primary command.

The tasks described in this section are:

- "Using primary commands with selection lists" on page 29
- "Scrolling to see data" on page 29

- "Sorting the data" on page 31
- "Finding specific data" on page 32
- "Changing the display width of columns" on page 32
- "Filtering your selection list" on page 33
- "Changing where (and if) columns are displayed" on page 34
- "Displaying data in hexadecimal" on page 34

Related topics
"REFRESH primary command" on page 804
"RESET primary command" on page 806

Using primary commands with selection lists

When you are viewing selection lists, you can enter primary commands on the Command line of the panel to perform these functions:

Action Command

Scroll the data
"BOTTOM primary command" on page 742
"DOWN primary command" on page 760
"LEFT primary command" on page 784
"RIGHT primary command" on page 810
"TOP primary command" on page 824
"UP primary command" on page 826

Scroll to a particular item in the list
"LOCATE primary command" on page 786

Find a particular item in the list
"FIND/FX primary command" on page 769
"RFIND primary command" on page 809

Tailor the displayed data
"RESET primary command" on page 806
"REFRESH primary command" on page 804
"TAILOR primary command" on page 823

Display data in hexadecimal format
"HEX primary command" on page 780

Related topics
Chapter 15, "Primary commands," on page 739

Scrolling to see data

To scroll through your data, use the following function keys:

Use this function key
To do this...
Scroll backward (up)
Up function key (F7)
Scroll forward (down)
Down function key (F8)
Scroll right
Right function key (F11)
Scroll left
Left function key (F10)

You can also scroll using the primary commands UP, DOWN, LEFT, RIGHT, TOP, and BOTTOM.

Examples

LEFT 8 Scrolls 8 columns to the left.

LEFT Scrolls left the number of columns indicated by the **Scroll** field.

LEFT CSR
If the cursor is positioned on a record, scrolls left to the cursor position; otherwise scrolls left one page.

UP 15 Scrolls up 15 lines.

UP Scrolls up the number of lines indicated by the **Scroll** field.

UP DATA
Scrolls up one line less than a page of data.

Related topics
"UP primary command" on page 826
"DOWN primary command" on page 760
"LEFT primary command" on page 784
"RIGHT primary command" on page 810
"TOP primary command" on page 824
"BOTTOM primary command" on page 742

Controlling how far you scroll: To control how far you scroll when you press one of the scrolling function keys, enter a scroll amount in the **Scroll** field:

Scroll amount
Scrolls...

PAGE One screen at a time

HALF Half a screen at a time

DATA One line or column less than a screen at a time

CSR To the cursor position (if the cursor is not positioned on a record when the scroll is performed, then the scroll amount defaults to PAGE)

MAX To the end of the data in the direction indicated by the scrolling function key.

nnnn A number of columns (when scrolling left or right) or a number of records or lines (when scrolling up or down) at a time

You can temporarily override the amount in the **Scroll** field by typing a scroll amount on the Command line, then pressing a scroll function key. For example, if you enter 8 on the Command line, then press the Right function key (F11), File Manager scrolls right 8 columns.

Scrolling to the first or last line or column: To scroll to the first or last line or first or last column in the selection list, type `MAX` (or `M`) on the Command line, then press one of the scroll function keys. For example, typing `M` then pressing the Right function key (F11) scrolls right to the last column.

You can also scroll to the first or last line by entering `TOP` or `BOTTOM` on the Command line.

Related topics

"TOP primary command" on page 824
"BOTTOM primary command" on page 742

Holding a column when scrolling left or right: You can set a "hold" option to hold specific columns. Held columns are displayed at the left of the screen and are always displayed regardless of your scrolling position or direction.

You can specify a column to be held in either of these ways:

- Set the column attributes for a *single* column:
  1. Scroll (if necessary) so that the column is displayed.
  2. Position the cursor within the column heading and press Enter to display a Column Details panel.
  3. Set the **Hold** value to `Y`.
  4. Press the Exit function key (F3) to return to the selection list.
- Set the column attributes for *one or more* columns:
  1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
  2. Set the **Hold** value to `Y` for each column you want to hold.
  3. Press the Exit function key (F3) to return to the selection list.

Related topics

"TAILOR primary command" on page 823

Sorting the data

You can change the order of the lines in a selection list by specifying the columns to be used in determining the sort order. For each column, you can specify a sort order of:

Ascending (A)
The contents of the column are sorted in ascending order. A down arrow (↓) is shown in the rightmost position of the column heading.

Descending (D)
The contents of the column are sorted in descending order. An up arrow (↑) is shown in the rightmost position of the column heading.

None (N)
The column is not used in determing the order of the lines.

Note: For any selection list, there is a default "held" column for which the sort order is, again by default, ascending (A).

You can specify a column to be used in determining the sort order in any of these ways:

- Setting the column attributes for a *single* column:
  1. Scroll (if necessary) so that the column is displayed.
  2. Perform any of these actions:
     a. Position the cursor within the column filter and press Enter to display a Column Settings panel.
     b. Set the **Sort** value to `A`, `D`, or `N` according to how you want the column sorted.
     c. Press the Exit function key (F3) to return to the selection list.

     or

a. Position the cursor within the column heading and press Enter. The column sort order is changed to the default. The default sort order for the column can be found in the Column Settings panel.

or

a. Tab *past* the column heading (to the rightmost position of the column heading area) and press Enter.
b. Each time you tab to this position and press Enter, the sort order rotates through ascending (↓), descending (↑), and none (no arrow displayed).

- Setting the column attributes for *one or more* columns:

  Perform either of these actions:

  1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
  2. Set the **Sort** value to `A`, `D`, or `N` according to how you want the column sorted.
  3. Press the Exit function key (F3) to return to the selection list.

  or

  1. Use the SORT command.

Note: When you change the sort order for a column, the sort order for all other columns is set to none.

Related topics
"TAILOR primary command" on page 823

Finding specific data

To find one or more occurrences of a character string in a selection list, you can use the FIND primary command, which you can abbreviate as F or /.

For example, to search for the next occurrence of the string "Apple" in any mix of uppercase and lowercase, enter this command on the Command line:

```
F APPLE
```

To find the next occurrence of the same string, use the RFIND command, or enter the FIND command with no argument. A message is displayed if the string cannot be found.

You can control the starting point, direction, and extent of the search with one of these optional parameters before or after the search string: NEXT, PREV, FIRST, LAST, ALL.

For example, to search for the last occurrence of the string "Orange", enter either of these commands on the Command line:

```
F ORANGE LAST
F LAST ORANGE
```

Related topics
"FIND/FX primary command" on page 769

Changing the display width of columns

You can change the display width of columns in a selection list

You can specify the display width of a columns in either of these ways:

- Set the column attributes for a *single* column:
  1. Scroll (if necessary) so that the column is displayed.

2. Position the cursor within the column heading and press Enter to display a Column Details panel.
3. Set the **Width** value to the number of characters you want to use to display the column.
4. Press the Exit function key (F3) to return to the selection list.

- Set the column attributes for *one or more* columns:
  1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
  2. Set the **Width** value to the number of characters you want to use to display each column
  3. Press the Exit function key (F3) to return to the selection list.

Related topics
"TAILOR primary command" on page 823

Filtering your selection list

You can restrict the data displayed in a selection list by specifying filters for any of the column data.

Points to consider when you specify a filter:

- Filtering is performed using a generic trailing match. That is, if the start of the data in the column being filtered matches the filter string, it is considered a match.

  For example, a filter of `PEA` matches PEA and PEAR, but not APPEAR.
- Matching of data to filters is not case-sensitive.

  For example, a filter of `PEA` matches PEA, PEAR, Pea and pear.
- You can specify any of these operators as the first character of the filter:

  `>` Greater than.
  `<` Less than.
  `=` Equal to.
  `!` Not equal to.

You can specify a filter in any of these ways:

- Set the column attributes for a *single* column:

  You can either:
  - Overtype the heading of the column you want to filter with a string.
  - Use the Column Details panel:
    1. Scroll (if necessary) so that the column is displayed.
    2. Position the cursor within the column heading and press Enter to display a Column Details panel.
    3. Type the filter in the **Filter** field.
    4. Press the Exit function key (F3) to return to the selection list.
- Set the column attributes for *one or more* columns:
  1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
  2. Type the filter in the **Filter** field for each column you want to filter.
  3. Press the Exit function key (F3) to return to the selection list.

Related topics
"TAILOR primary command" on page 823

Changing where (and if) columns are displayed

You can change the order that data columns are displayed in a selection list. You can also suppress columns from being displayed.

To change the order in which data columns in a selection list are displayed:

1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
2. Set the **Order** value for each column according to the order (left to right) that you want the columns displayed. The values you specify must be unique to ensure that the columns are re-ordered as intended.
3. Press the Exit function key (F3) to return to the selection list.

To suppress a data column from being displayed in a selection list are displayed:

1. Enter the TAILOR primary command to display a list of columns together with their column attributes.
2. For any column that you want to suppress from being displayed in the selection list, set the **Order** value to 0 (zero). Note that columns whose **Order** value is set to 0, are displayed after the other (ordered) columns.
3. Press the Exit function key (F3) to return to the selection list.

Related topics

"TAILOR primary command" on page 823

Displaying data in hexadecimal

To display the data in a selection list in hexadecimal format, use the HEX primary command.

Related topics

"HEX primary command" on page 780

Selecting a range of PDS(E) members

The following File Manager entry panels include the option, **Advanced member selection**:

- "Print Utility panel" on page 620
- "Find/Change Utility panel" on page 574
- "Copy From panel" on page 491
- "Template Build Utility panel" on page 688
- "Load Module Compare - entry panel" on page 590
- "Load Module Information panel" on page 592
- "Compare Utility: "Old" and "New" panels" on page 471

The **Advanced member selection** option lets you select a range of PDS(E) members based on one or more of the following criteria:

- Similar names
- The user ID by which they were last updated
- The date they were created
- The date they were last modified

To select a range of PDS(E) members in this way, select the **Advanced member selection** option on the relevant entry panel and press Enter.

File Manager displays the Advanced Member Selection panel:

Member name

To select a range of PDS(E) members based on their name, perform either of these actions

- Specify the required member name (or a mask) in the **Member name** field.
- To select a range of members with similar names, enter the "from" and "to" values in the **from** and **to** member name fields respectively.

  If you omit the **from** member name value, all members from the beginning of the data set to the member name **to** value are included. File Manager pads the field with blanks to 8 characters.

  If you omit the **to** member name value, all members from the member name **from** value to the end of the data set are included. File Manager pads the field with blanks to 8 characters. If you enter an asterisk as the last character in the member name **to** field, File Manager pads the field with high values to 8 characters.

  For example, the following Member name range selects all members with a name beginning with D, E, or F:

```
or range from: D
           to: F*
```

User ID

To select a range of PDS(E) members based on the user ID by which they were last updated, perform either of these actions

- Specify the required user ID (or a mask) in the **Updated by** field.
- To select a range of user IDs, enter the "from" and "to" values in the **from** and **to** user ID fields respectively.

  If you omit the **from** user ID value, all members from the beginning of the data set to the user ID **to** value are included. File Manager pads the field with blanks to 7 characters

  If you omit the **to** user ID value, all members from the user ID **from** value to the end of the data set are included. File Manager pads the field with blanks to 7 characters. If you enter an asterisk as the last character in the user ID **to** field, File Manager pads the field with high values to 7 characters.

  For example, the following user ID range selects all members with a user ID (by which they were last updated) in the alphabetic range "GEOFF" to "GEORGE":

```
from: GEOFF
  to: GEORGE
```

Date created

To select a range of PDS(E) members based on the date they were created, perform either of these actions

- Specify the required creation date in YYYY/MM/DD format (or a mask) in the **Date created** field. You can specify an asterisk (*) as the last character to indicate a range of dates, or a percent sign (%) in place of a single character to indicate a selection of dates.
- To select a range of creation dates:
  - Enter the "from" value (in YYYY/MM/DD format) in the **from** creation date field. If you omit the **from** creation date, or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
    DD = 01
    MM = 01

YYYY = 0000

No other wildcarding is allowed.

– Enter the "to" value (in YYYY/MM/DD format) in the **to** creation date field. If you omit the **to** creation date, or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

For example, the following creation date range selects all members created between 2004/07/05 and 2004/12/31:

```
from: 2004/07/05
  to: 2004*
```

Date modified

To select a range of PDS(E) members based on the date they were last modified, perform either of these actions

- Specify the required modification date in YYYY/MM/DD format (or a mask) in the **Date modified** field. You can specify an asterisk (*) as the last character to indicate a range of dates, or a percent sign (%) in place of a single character to indicate a selection of dates.
- To select a range of modification dates:
  – Enter the "from" value (in YYYY/MM/DD format) in the **from** modification date field. If you omit the **from** modification date, or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
  DD = 01
  MM = 01
  YYYY = 0000

  No other wildcarding is allowed.
  – Enter the "to" value (in YYYY/MM/DD format) in the **to** modification date field. If you omit the **to** modification date, or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
  DD = 31
  MM = 12
  YYYY = 9999

  No other wildcarding is allowed.

For example, the following modification date range selects all members that were last updated between 2003/11/01 and 2003/12/31:

```
from: 2003/11*
  to: 2003
```

Related topics

"Advanced Member Selection panel" on page 439
"Printing from File Manager" on page 299
"Finding and changing data in multiple PDS members" on page 270
"Copying data sets" on page 253
"Print Utility panel" on page 620
"Find/Change Utility panel" on page 574

"Copy From panel" on page 491
"Template Build Utility panel" on page 688
"SELECT primary command" on page 814
"Member Selection panel" on page 600

Using hexadecimal values in File Manager

In many File Manager panels, you can view and enter decimal values as text characters or as hexadecimal values. To help you quickly convert decimal values to hexadecimal or hexadecimal values to decimal values, File Manager provides a simple converter tool, in the form of two primary commands.

To convert a decimal value to its equivalent hexadecimal value:

1. On the Command line of any panel, enter DX followed by the decimal value.
   File Manager displays the results in a message at the bottom of your screen.

For example, if you enter the following on a File Manager Command line:

```
DX 10
```

then File Manager displays this message box:

```
Dec 10 = hex 0000000A
```

To convert a hexadecimal value to its equivalent decimal value:

1. On the Command line of any panel, enter XD followed by the hexadecimal value.
   File Manager displays the results in a message at the bottom of your screen.

For example, if you enter the following on a File Manager Command line:

```
XD 10
```

then File Manager displays this message box:

```
Hex 00000010 = dec 16
```

Related topics

"DX primary command" on page 761
"XD primary command" on page 829
"Selecting a display format" on page 66

Getting help

File Manager offers two levels of help information: context-sensitive field information and Tutorial Help. The field help information provides a description of your current field, lists the valid values or range of values that can be entered, and gives the default field value where applicable. Tutorial Help is a context-sensitive help system that provides detailed explanations of the File Manager panels and the processes in which they are used. In addition, all error messages have an associated full text explanation, which can help you to determine the cause of a problem.

The tasks described in this section are:

"Displaying field help information" on page 38

"Accessing Tutorial Help"
"Viewing error messages" on page 40

Displaying field help information

To display the field help information:

1. Place your cursor on the field entry line.
2. Press the Help function key (F1). The field help information is displayed in a pop-up window.

   If further information is needed, press the Extended Help function key (F5), to access the Tutorial Help page for the current panel.

   Pressing Help while the field information window is displayed takes you to the Help Tutorial page within the Tutorial Help system (that is, the page that describes how to use Tutorial Help).
3. When you have finished reading the message or the relevant Tutorial Help page, press the Exit function key (F3) to return to your starting panel.

For example, Figure 15 shows the message seen when F1 is pressed from the RECLIMIT field in the Set Print Processing Options panel.

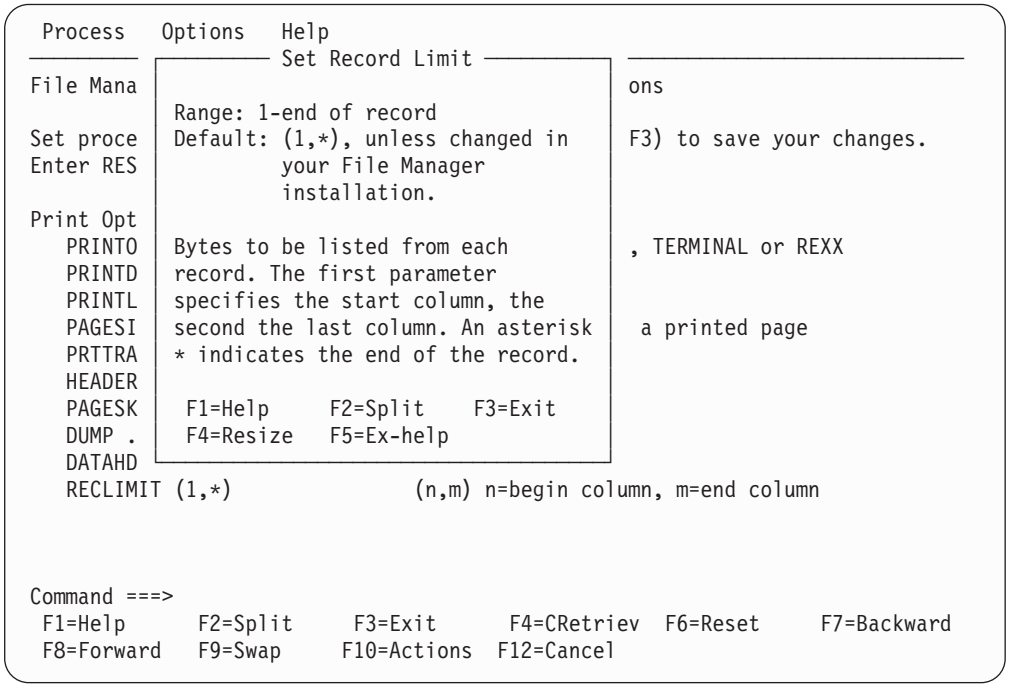


Figure 15. Field help information for the RECLIMIT field

Accessing Tutorial Help

The Tutorial Help system is both context-sensitive and structurally organized. You can get directly to the information you need, using the context-sensitive access method or you can enter the Help system at a specific location and navigate to any topic information within the structure.

To access a context-sensitive Tutorial Help page:

1. Place your cursor on the Command Line or anywhere in the panel that is outside of a field entry line.
2. Press F1. The Tutorial Help page associated with the current panel is displayed.

When you access the Tutorial Help in this way, the first page that is displayed may be a main topic page, with a menu listing the associated sub-topics, or it may be a sub-topic page, depending upon the context from which you started.

3. Navigate through the Tutorial Help pages, to find the information you require (see below for a list of navigation commands).
4. When you have finished, press F3 (Exit) to return to your starting panel.

The Tutorial Help system is organized with a Table of Contents structure that is based upon the File Manager Primary Options menu. Each entry within the Table of Contents leads to a main topic with a number of associated sub-topics. In addition, there is an index that lists selected topics from the Tutorial Help, alphabetically by subject.

To choose your starting point in Tutorial Help:

1. Select the Help pull-down menu from any panel Action Bar.
2. Enter the option number for the Tutorial Help entry point that you require. These are:

   1. Help for help
   Displays the Help panel for the Tutorial Help system.

   2. Extended help
   Displays the Tutorial Help panel associated with the current File Manager panel (equivalent to pressing F1 from the File Manager panel). When on the Primary Options menu, this is the Tutorial Help Table of Contents panel.

   3. Keys help
   Displays a panel that provides help for the Function Keys that are active on the current File Manager panel.

   4. Help index
   Displays the Help Index.

   5. Tutorial
   Displays the Tutorial Help Table of Contents panel.

   6. About
   Displays the File Manager version and release information in a pop-up window.

   7. News about File Manager
   Displays a panels providing general information about the current File Manager version/release.

To navigate Tutorial Help, enter one of the following commands in the Command line on any Tutorial page:

BACK or B
To back up to the previously viewed page.

SKIP or S
To skip the current topic and go on to the next topic.

UP or U
To display a higher level list of topics.

TOC or T
To display the Table of Contents.

INDEX or I
To display the Tutorial Index. Once the Index has been displayed, use the RIGHT (F11) and LEFT (F10) keys to scroll through the list, then position your cursor on a subject and press ENTER (Ctrl) to display the topic.

Alternatively, you may use the following keys whenever you are in the tutorial:

ENTER (Ctrl)
To display the next sequential page within a topic.

HELP (F1)
To display the Help Tutorial page within the Tutorial Help system (that is, the page that describes how to use Tutorial Help).

END (F3)
To terminate the tutorial.

UP (F7)
To display a higher level list of topics (instead of typing UP).

DOWN (F8)
To skip to the next topic (instead of typing SKIP).

RIGHT (F11)
To display the next sequential page within a topic (instead of pressing ENTER).

LEFT (F10)
To display the previous sequential page within a topic (instead of typing BACK).

Note: The listed keys are the default key mappings. As you can customize key mappings, they may be different on your system.

Viewing error messages

If an error occurs when File Manager attempts to process a panel, a short text message displays in the upper right corner of the screen. While this message is displayed, pressing F1 displays the expanded text of the error message in either a box at the bottom of the screen, or underneath the command line, dependant upon your setting of the ISPF option "Long message in pop-up". The following example is shown with that ISPF option set on.

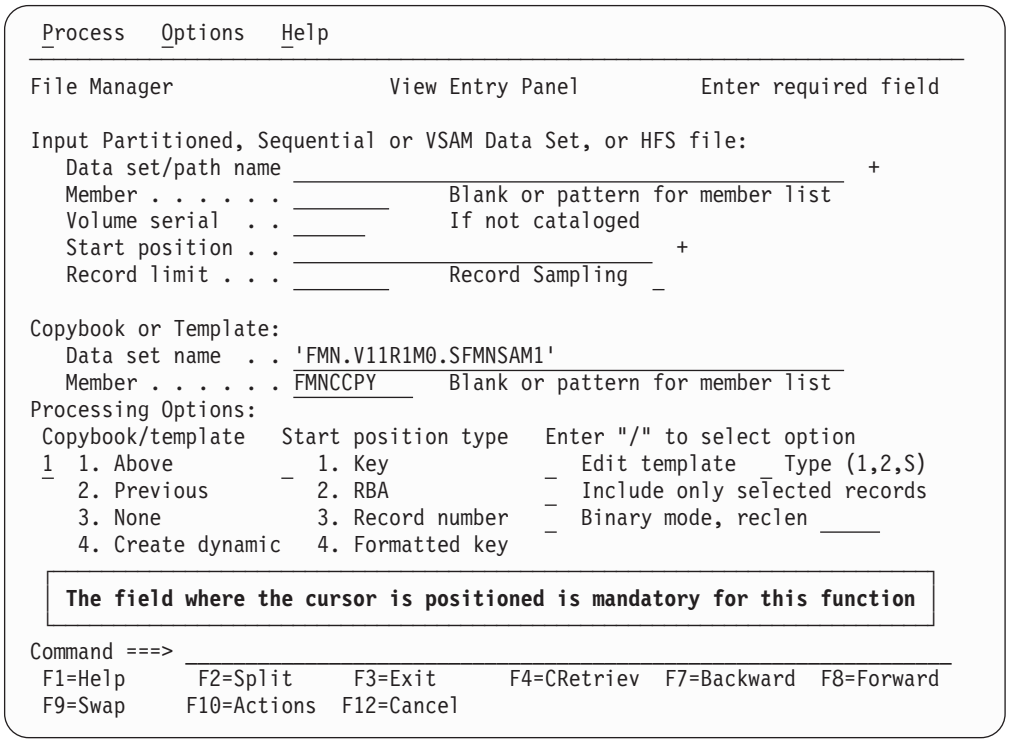


Figure 16. View Entry panel showing expanded message

Related topics

"Messages" on page 1167

Setting your default processing options

Many of the processing operations performed by File Manager utilize default values that can be set from within the File Manager application. By adjusting these values, you can customize File Manager so that its behavior is best suited to your needs. Your settings for these options are stored in your ISPF profile, and are invoked when you log in, regardless of which workstation you use.

You can update these default values by accessing the relevant processing option panel.

To access a processing option panel, use any one of the following methods:

- From the File Manager Primary Options Menu:
  1. Select Option 0 (Settings). The Set Processing Options Menu panel is displayed.
  2. Select the required processing options type from the menu.

```
 Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────────
 File Manager                Set Processing Options

 1  Print     Print settings
 2  System    System settings
 3  Tape      Tape processing options
 4  Batch     Job card specifications
 5  Compiler  Language and compiler specifications
 6  EDIT      Editor options
 7  Sharing   VSAM Edit sharing options
 8  Temporary Temporary Data Set Allocations
 9  Output    Output Data Set Allocations
 10 Trace     Trace options




 Command ===> _________________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 17. Set Processing Options menu

- From any File Manager panel, use the **Options** pull-down menu to select the required processing options type:

```
  Process    Options   Help
 ────────  ┌───────────────────────────────────────┐  ────────────────────────────
 File Man  │ __ 1. Print settings                  │
           │    2. System settings                 │
 1  Print  │    3. Tape processing options         │
 2  Syste  │    4. Job card specifications         │
 3  Tape   │    5. Compiler language selection     │
 4  Batch  │    6. COBOL compiler specifications   │
 5  Compi  │    7. HLASM compiler specifications   │
 6  EDIT   │    8. PL/I compiler specifications    │
 7  Shari  │    9. Editor options                  │
 8  Tempo  │   10. VSAM Edit sharing options       │
 9  Outpu  │   11. Temporary Data Set Allocations  │
 10 Trace  │   12. Output Data Set Allocations     │
           │   13. Trace options                   │
           │   14. ISPF settings                   │
           └───────────────────────────────────────┘




 Command ===>
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 18. Options pull-down menu

- On the Command/Options line of any File Manager panel, enter an equals sign (=), followed by the option number for the required processing type. For example, to display the Set Tape Processing Options panel, enter `=0.3`.

When you select an option processing type, File Manager displays a panel listing those options and their current values. The following notes apply to all of these option panels:

- To change the value of an option, overtype its current value.
- To save the options you have changed and exit the panel, press the Exit function key (F3).

Changes are saved in your ISPF user profile for future File Manager sessions. They remain in effect until you change the option again.

- To exit a panel without saving changes, press the Cancel function key (F12).
- To reset all options on the panel to their installation defaults, enter RESET on the Command line.
- The options on these panels only affect the behavior of File Manager panels. To set options when programming with File Manager functions, use the equivalent function keywords. You can set most options using the SET function (see "SET (Set Processing Options)" on page 1047). For Compiler Language Selection, COBOL, PL/I or HLASM options (for compiling copybooks into templates), use the keywords on the same function where you specify the copybook (the DSEB, DSC, DSG, DSM, DSP or DSU functions).

The use of most of these processing options is discussed in the section that the option affects. For example, the use of Set Print Processing Options panel is described in the **Printing from File Manager** section of the **Managing data sets** chapter. Only those processing options that affect more than one area of File Manager's functionality are described in this section.

Related topics

"Set Processing Options panel" on page 661

Set print processing options (option 0.1)

The print processing options control the print output that the File Manager functions produce.

Related topics

"Setting your Print Processing Options" on page 299
"Set Print Processing Options panel (option 0.1)" on page 658

Setting your System Processing Options (option 0.2)

The System Processing Options control such things as data presentation and conversion values, and whether or not user-written I/O exits are allowed when processing various utility functions.

In the Data presentation and conversion section:

- The PAD option defines both the padding and truncation of fields within records when performing actions such as copying or creating data sets.
- The NOTRUNC option specifies that, if the PAD option has been selected, no truncation is performed when copying or writing records to a variable length data set.

The CYLHD option defines how File Manager interprets cylinder-head and track-number values in disk functions. ABSOLUTE means the actual physical addresses. RELATIVE means relative to the start of the data set.

The EOD option sets the end-of-data delimiter for tape input.

The CCSID option specifies the CCSID to be used when translating fields which have a associated CCSID when File Manager is run in batch.

The User I/O Exit Specification section contains two fields that, together, control whether or not you are able to use custom user I/O exits in tasks such as viewing, editing, printing, copying or comparing data sets. User I/O exits are

custom-written programs that handle files that require pre- or post-processing not offered directly by File Manager. This may include handling data compression or encryption, or other non-standard activities performed on the data set records. There are no restrictions on the programming language that you can use to write an exit, however, the exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

To allow the use of user I/O exits, several things must occur:

1. When File Manager is installed, the person doing the initial customization (the FM Administrator) must have chosen to allow user I/O exits on a site-wide basis.

   If the FM Administrator chose not to allow user I/O exits, the **Exit enabled** field is set to **NO Disabled by installation options**. You cannot change this from within the File Manager panel and you cannot use I/O exits in online or batch processing.

   If the FM Administrator chose to allow user I/O exits, you can then set the **Exit enabled** field to YES. Choosing YES means that when performing the following tasks (via a panel or function), you have the option to use or not use a user I/O exit:

   - Viewing data sets (View panel only)
   - Editing data sets (Edit panel, DSEB and DSU functions)
   - Finding and changing data (Find/Change panel, FCH function)
   - Copying data sets (Copy From and Copy To panels, DSC function)
   - Creating data (Data Create panel, DSG function)
   - Comparing data sets (Old and New panels, DSM function)
   - Printing data sets (Print Utility panel, DSP function)

   Choosing NO means that you are not able to select an exit in any of the File Manager panels. However, you are still able to specify a user I/O exit in batch processing.
2. Either you or someone in your organization must have developed your own custom exit to be used. Sample programs and control block templates have been provided for HLASM (High Level Assembler), COBOL, and PL/I but these need to be tailored to your site needs.

   If the FM Administrator has allowed user I/O exits, he or she may have provided a default exit for you to use. If they have done so, the name is displayed in the **Default exit** field.

   You can override this system default with your own default exit, by changing the name in this field. In each applicable panel or function, you can then choose to accept or override the current default.
3. When you want to use a user I/O exit, regardless of whether you are using the system default, your own default or specifying a unique exit at the individual panel or function level, the name of the exit must be the name of a PDS(E) member of a data set that is included in the current STEPLIB/ISPLLIB concatenation for File Manager.

Related topics

"Set System Processing Options panel (option 0.2)" on page 663

Set tape processing options (option 0.3)

The tape processing options control such things as the type of labels to be created and the type of data translation to be used when manipulating data on a tape.

See “Set Tape Processing Options panel (option 0.3)” on page 665 for panel details.

Set batch job card information (option 0.4)

On the Batch Job Card Information panel, you can specify batch job card information to be used for generating batch job submission JCL. The Batch Job Card Information panel provides a number of blank lines (**Batch Submission Job Statement Information**) where you can enter this information.

File Manager assumes any non-blank lines define a JCL job card and copies the information from these lines into generated JCL. If all the lines are blank, the JCL generation process generates a basic job card using information from system variables.

Related topics

“Setting processing options in batch” on page 402
“Set Batch Job Card Information panel (option 0.4)” on page 649

Language and compiler specifications (option 0.5)

The language and compiler specifications allow you to specify the compiler that File Manager uses to build templates, and also to set options for that compiler.

Related topics

“Setting your template processing options” on page 140
“Compiler Language Selection panel” on page 470
“Set COBOL Processing Options panel (option 0.5.2)” on page 649
“Set PL/I Processing Options panel (option 0.5.4)” on page 656
“Set HLASM Processing Options panel (option 0.5.3)” on page 652
“Set Language and Compiler Specifications (option 0.5)” on page 654

Editor options (option 0.6)

The editor options control various options used to format the display or print of data when viewing, editing, or printing.

In an editor session, use the PROFILE primary command to display the current settings.

Note: If you are viewing, editing, or printing data, the **Options** pull-down menu choice to change these options is not available. The options may only be changed before starting your View, Edit, or Print session. To change the behavior for the current View, Edit, or Print session, use an equivalent primary command (where available).

Related topics

“Editor Options panel” on page 546
“Displaying your current editor options” on page 59
“PROFILE primary command” on page 800

VSAM edit sharing options (option 0.7)

The VSAM edit sharing options control the behaviour when a file is being shared by other users.

See “VSAM Edit Sharing Options panel” on page 721 for panel details.

Temporary Data Set Allocations (option 0.8)

The Temporary Data Set Allocations options control the allocation defaults for temporary and auxiliary data sets.

See "Set Temporary Data Set Allocation Options panel" on page 667 for panel details.

Output Data Set Allocations (option 0.9)

The Output Data Set Allocations options control the allocations defaults for output data sets.

See "Set Output Data Set Allocation Options panel" on page 655 for panel details.

Trace options (option 0.10)

The trace options control the Trace output produced by File Manager when it is run in debug mode.

See "Set Trace options panel" on page 669 for panel details.

Displaying or editing DBCS characters

If you want to display or edit DBCS characters in File Manager and you have a DBCS capable terminal, you must ensure that your ISPF settings specify a terminal type that supports these characters.

To specify the Terminal Type in ISPF:

1. Exit from File Manager and return to the z/OS Primary Option Menu panel.

   Although you can change the ISPF Terminal Type setting while File Manager is running, the change is not reflected until the next time that File Manager is invoked.
2. Select option `0. Settings`. The ISPF Settings panel is displayed.

```
 Log/List  Function keys  Colors  Environ  Workstation  Identifier  Help
 ──────────────────────────────────────────────────────────────────────────
                              ISPF Settings
                                                              More:   -
   Session Manager mode                   Command delimiter . ;
 / Jump from leader dots
   Edit PRINTDS Command
 / Always show split line
   Enable EURO sign

Terminal Characteristics
  Screen format   1  1. Data    2. Std     3. Max     4. Part

  Terminal Type   3    1. 3277       2. 3277A      3. 3278       4. 3278A
                       5. 3290A      6. 3278T      7. 3278CF     8. 3277KN
                       9. 3278KN    10. 3278AR    11. 3278CY    12. 3278HN
                      13. 3278HO    14. 3278IS    15. 3278L2    16. BE163
                      17. BE190     18. 3278TH    19. 3278CU    20. DEU78
                      21. DEU78A    22. DEU90A    23. SW116     24. SW131
                      25. SW500

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=Actions  F12=Cancel
```

Figure 19. ISPF Settings panel

3. Select either 3277KN or 3278KN from the list of Terminal Types.

Tip: You may need to scroll down this panel to find the list of Terminal Types.

4. Press Exit (F3) to save your selection and return to the z/OS Primary Option Menu panel.
5. Restart File Manager.

National characters

File Manager uses the national characters shown in Table 2.

Table 2. National characters

Character	Hexadecimal value	Displayed as, in code pages 37 and 500
Dollar sign	X'5B'	$
Pound sign	X'7B'	#
At sign	X'7C'	@

Note:

1. The dollar sign ($) and the pound sign (#) have special syntactical meaning in File Manager syntax.
2. In countries using code pages other than 37 and 500:
   - The above characters as represented on terminal keyboards might generate a different hexadecimal representation, and this might cause an error or unwanted results. For example, in some countries the $ character might generate a X'4A'.
   - The above hexadecimal values might display as different characters to those shown.

When you enter File Manager commands in batch or online, use the keyboard characters that correspond to the hexadecimal values shown in Table 2.

SAF-rule controlled auditing

The way in which auditing is performed during a File Manager session may be determined by SAF rules, depending on the auditing requirements of your installation.

SAF rules for auditing

When auditing is determined by SAF rules:

- Auditing can be specified to either your audit log data set, to SMF, or to both destinations.
- All File Manager functions which access data sets and MQ data may be subject to audit.
- It determines if the **Create audit trail** option is displayed on the Edit Entry panel and the WebSphere® MQ Queue Editor Entry panel.

For the File Manager functions which SAF-rule controlled auditing may be applied to, the SAF rules can be specified at a resource or function level allowing auditing at the FUNCTION, UPDATE, or ALL level. Auditing at the ALL level indicates that record reads and updates are recorded, UPDATE indicates only inserted, deleted, or changed records are recorded, while FUNCTION records only the event.

SAF-rule controlled auditing not in effect

When SAF-rule controlled auditing is not in effect (that is, auditing is controlled by the settings in the FMN0POPT macro), the **Create audit trail** option is displayed on the Edit Entry panel and the WebSphere MQ Queue Editor Entry panel. If the

Create audit trail option is selected, or the AUDITLOG option DEMAND or YES has been specified in the FMN0POPT macro, auditing of inserted, deleted, or changed records are logged.

SAF-rule controlled auditing in effect, auditing possible

When SAF-rule controlled auditing is in effect, and auditing is to occur, the actual decision on whether or not to create an audit log depends on the SAF rules in place.

If you have control-level audit access, the **Create audit trail** option is displayed on the Edit Entry Panel and the WebSphere MQ Queue Editor Entry panel. If you select this option, and no other SAF rule is in place for this function and resource, auditing of inserted, deleted, or changed records is logged. If there is an existing SAF rule for this function and resource, the level of audit specified in the SAF rule applies.

Chapter 3. Viewing and changing data sets

To view data stored in a supported data set, use the View Utility or the Edit Utility:

View Utility

The View Utility (option 1) allows you to view and temporarily change the data, but without the ability to save any changes to the data set being viewed. Changes may be saved to a different data set.

When you initiate the File Manager editor with the View Utility, this is referred to as being in a "View editor session" (or just "View").

The view function, similar to ISPF view, provides an editor session that works essentially the same as the Edit Utility, but without the need for exclusive access to the data set. It presents you with an interface that is capable of holding your temporary changes to data. This may be useful, for example, for assessing a change but without any risk of altering live data.

Note: If you have changed the data in a View editor session, even though you are not able to save the changes back to the same data set you are viewing, you *can* save the changed data to *another* data set (either existing or new, however MQ queues are not supported by these commands) by issuing one of these primary commands:

APPEND

Appends specified records in the current editor session to another (existing) data set.

CREATE

Creates another (new) member or data set from specified records in the current editor session.

REPLACE

Replaces another (existing) member or data set with specified records in the current editor session.

SAVEAS

Saves the currently exposed data in the current editor session to another (new) member or data set, and switches to an edit of the nominated member or data set.

Related topics

"Creating, replacing, and adding data using existing data" on page 246

The first time you make a change to the data in a View editor session, File Manager displays a warning as shown in Figure 20 on page 50.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
View             FMNUSER.TEST.DATA(EMP)                              Top of 40
                                                 Record AT TOP        Format TABL
       REC-TYPE NAME                 EMPLOYEE-NO     AGE    SALARY    MONTH(1)
       #2       #3                            #4      #5        #6          #7
       AN 1:2   AN 3:20                  BI 23:2 BI 25:2  PD 27:4      BI 31:4
       <>       <---+----1----+---->      <---+>  <---+> <---+--> <---+----1>
****** ****  Top of data  ****
000001 01       Grant Sutherland            7712      56    75111            6
000002 01       Andrew Estle                6612      21    75000           30
000003 01       Graham Purdie               5512      24    68000            7
000004 01       Bill Soper                  4523      28    68000            5
000005 01       Tyrone Dalais               3170      21    60000           14
000006 01       Rod Turner                  5006      35   100000           28
000007 01       Clive Nealon                2257      47   100000           44

 Save is not possible in a View session. Use Edit if you want to be able to
 save changes.

Command ===>                                                        Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 20. Warning pop-up displayed after first change in View editor session

When you exit from a View editor session by pressing the Exit function key (F3), if you have made a change to the data in the editor session, File Manager displays a warning as shown in Figure 21.

```
 Process   Options   Help
 - ---------------------- View Warning ----------------------   ------------
V                                                               Top of 40
            You are currently in Data View mode:                 Format TABL
                                                                   MONTH(1)
   Press Enter to confirm exit from View.  No changes will be            #7
   saved.                                                           BI 31:4
                                                                <---+----1>
*  Enter the END or EXIT command to return to View, where you
0  can use the  CREATE or REPLACE commands to save your                   6
0  changes.                                                              30
0  Command ===>                                                           7
0   F1=Help       F2=Split      F3=Exit       F7=Backward                 5
0   F8=Forward    F9=Swap      F12=Cancel                                14
0                                                                        28
000007 01       Clive Nealon                2257      47   100000           44
000008 01       Jim Alexander               1332      63   125000           47
000009 01       Silvano Prez                2895      50    60000            1
000010 01       Don Pharoah                 4890      55    63000           38
000011 01       John Levrington             3344      40    67000           14
000012 01       Liz Rushton                 3789      40    66000           44
Command ===>                                                        Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 21. Warning pop-up displayed when exiting from a View editor session if changes made to data

Edit Utility

The Edit Utility (option 2) allows you to view and change your data (by inserting, deleting and otherwise modifying records), and save any changes you make.

When you initiate the File Manager editor with the Edit Utility, this is referred to as being in an "Edit editor session" (or just "Edit").

These instructions are given within the context of the File Manager editor. In most cases, you can perform these actions in either Edit or View, in which case the term "editor" is used. Similarly, the term "editor session" can indicate either a View editor session or an Edit editor session.

Many of the tasks described in this chapter require the use of templates. For information on how to define and edit templates, see Chapter 4, “Creating and editing templates,” on page 127.

The major tasks described in this chapter are:

Starting and ending editor sessions

The File Manager editor allows you to edit your data as logical records, in an unformatted or formatted state. Records that are structured into fields are particularly suitable for editing in a formatted state using a template. Data that is relatively unstructured, such as a text file, copybook definition or REXX procedure, can be edited in its unformatted state.

If your data has been written using the ISPF PACK option, you can choose to unPACK the data for easy editing or edit the data in its PACKed format. You can also choose to write previously unPACKed data to the ISPF PACK format. Packing only applies to physical sequential data sets or to members of a PDS(E).

The tasks described in this section are:

Starting an editor session without using templates

To display your data in an unformatted state:

1. From the File Manager Primary Options Menu panel, select one of these:
   1 View
   Displays the View Entry panel.

   Note: This option does not allow you to save any changes you make to the data.
   2 Edit Displays the Edit Entry panel.
2. Specify your HFS file or data set name or name pattern, member name or pattern and/or volume serial.
3. If required, specify a **Starting position** for the data, then select a **Start position type** from the Processing Options.

When your data set is a KSDS file, you can specify a key of up to 250 characters, as the starting position. To make it easier to enter a key of this size, you can expand the field by pressing the EXPAND function key (F4) or you can scroll within the field by pressing the LEFT or RIGHT function keys (F10/F11).

When using a record number as the start position, you can specify a negative number to start from the given number of records before the end of the data set. The initial display is positioned at the specified record. Records prior to your starting position are not visible.

Note: See also the use of the KEY primary command in step 9.

4. If required, specify a record limit. The count of displayed records commences from the Starting position (which defaults to the top of the data set).

   You can specify a record limit as the word "MEMORY" so that File Manager loads only as many records as will fit comfortably into storage.

5. If required, select the **Record Sampling** field. Selecting this field will cause the Record Sampling panel to display after the Entry panel has been processed.

   You can use record sampling to specify a pattern for reading records from the data set. In your pattern, you can specify a starting position, read a number of records, skip a number of records, and then repeat the pattern until an end point is reached. The end point might be a specified number of physical records, a specified number of records selected by a template, the end of the input file, or the maximum number of records that can safely fit within the available virtual storage (whichever is reached first).

   When record sampling is in effect:

   - The values (if any) in the **Start position** and **Record limit** fields are carried through into the Record Sampling panel.
   - The skipped records are not presented or accessible within the session.
   - The session will run in In-place mode and will only contain as many records as will fit into the available virtual storage.

6. In Edit, if you only want to update records in the data set (without inserting or deleting records), then select the **Inplace edit** option.

   Note that you cannot insert or delete records when:
   - **Record Sampling** is selected,
   - A **Record limit** of "MEMORY" is specified,
   - The data set is a VSAM file defined as NOREUSE, or
   - A template is specified which indicates segmented records
   - The data set is a QSAM LRECL=X that is too large to be contained in memory

7. Type 3 (None) in the **Copybook/template usage** field.

   Note: When this field is set to None, entries in the **Copybook or Template Data set name** and **Member** fields are ignored.

8. In Edit, if required, select the **Create audit trail** option.

   Note: To print an audit report, use Print audit trail report (see "Edit Entry panel" on page 541).

9. Press Enter to process the panel.

   If you have provided the complete name of your data set and data set member, the Edit (or View) panel displays your data.

   If you used a pattern to specify your data set or member, the Data Set Selection or Member Selection panel is displayed. Select the required data set or

members from the list by typing an S in the Prefix field, then pressing Enter. The Edit (or View) panel displays your data.

If your data set is a KSDS file, you can specify a position to jump to by issuing the KEY primary command.

Related topics

"Specifying a data set and a member name" on page 16
"Scrollable input and display fields for long names" on page 80
"View Entry panel" on page 713
"View panel" on page 707
"Record Sampling panel" on page 628
"KEY primary command" on page 783

Example of displaying data without a template

Follow the steps in this example, to display a data set member that contains a copybook definition (suitable for viewing without a template).

1. From the File Manager Primary Option Menu panel, select **1 View**.
2. In the **Data set name** field, type 'FMN.V11R1M0.SFMNSAM1'.
3. In the **Member** field, type FMNCCPY.
4. Type 3 (None) in the **Copybook/template usage** field.
5. Press Enter to process the panel. Your results should look similar to below:

```
 Process   Options   Help
--------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCCPY)                    Top of 34
       Col 1       Insert Length 80          Record AT TOP     Format CHAR
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001       *** ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
000002       *   IBM File Manager for z/OS Version 11
000003       *   Licensed Materials - Property of IBM
000004       *
000005       *   5655-W47
000006       *
000007       *   (c) Copyright IBM Corp. 1986, 2010. All Rights Reserved.
000008       *
000009       *   US Government Users Restricted Rights - Use,
000010       *   duplication or disclosure restricted by GSA ADP
000011       *   Schedule Contract with IBM Corp.
000012       *
000013       *   Sample COBOL copybook to demonstrate the File Manager
000014       *   field level processing.
000015       *   The associated template is FMNCTMPL and the sample
Command ===>                                                    Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 22. Viewing without a template

6. Press F8 to scroll down and view the copybook contents.
7. Press F3 to exit and return to the Edit (or View) Entry panel.

Starting an editor session with a template

To display your data with a template:

Note: This process assumes that you have an existing template that you can use to format your data. For details on creating new templates or editing an existing template before use, see Chapter 4, "Creating and editing templates," on page 127.

1. Follow steps 1 to 6 in "Starting an editor session without using templates" on page 51.
2. If required, select the **Include only selected records** option. This option is only effective when using a template and you are using either:
   - Record sampling, or
   - A record limit of "MEMORY"

   Note: If you have selected the **Include only selected records** option, the editor session operates in an in-storage way.
3. In the **Copybook or Template** section of the panel, specify the data set in which your copybook template or dynamic template is stored, or use a wildcard character to select from a list.

   Note: The Use I/O exit option only affects the data set that you are choosing to view. It cannot be used to process the template or copybook that you are using to format the data.
4. In the same section, specify the Member name or leave the Member name field blank to choose from the list of Members in a partitioned data set.
5. In the **Copybook/template usage** field, type **1** to select the Above option.

   Alternatively, if you have previously used a template with the input data set, you can select option **2. Previous**. The Previous option instructs File Manager to ignore the contents of the **Copybook or Template** fields, and instead use the template that you have most recently associated with the input data set.

   If you want to edit the template before using it, select the **Edit template** field.
6. Press Enter to process the panel.

   If you have provided the complete name of your input data set and data set member and your template data set and data set member, the Edit (or View) panel displays your data.

   If you used a pattern to specify your data sets or members, the Data Set Selection or Member Selection panels are displayed in the following order:
   - Data Set Selection panel
   - Template Data Set Selection panel
   - Template Member Selection panel
   - Member Selection panel

   Select the required data sets or members from the lists by typing an S in the Prefix field, then pressing Enter. The Edit (or View) panel displays your data.

Related topics

Chapter 4, "Creating and editing templates," on page 127
"View Entry panel" on page 713
"View panel" on page 707
"Specifying a data set and a member name" on page 16

Example of displaying data with a template

The first task in this tutorial is to view data that has been structured into fields, without using a template. This demonstrates some of the display issues that can occur when unformatted data is viewed. In the next task, you view the same data, with a template.

1. From the File Manager Primary Option menu, select option **1. View**.

2. In the **Data set name** field for the input data set, type `'FMN.V11R1M0.SFMNSAM1'`.
3. In the **Member** field for the input data set, type `FMNCDATA`.
4. In the **Copybook/template usage** field, select option **3. None**.
5. Press Enter.

   Figure 23 shows what this data looks like without a template.

```
 Process   Options   Help
_____________________________________________________________________________
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                  Top of 40
       Col 1_____ Insert Length 80_____          Record AT TOP___ Format CHAR
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 01Grant Smith         ..............................................
000002 01Andrew Apple        .5....&.......................................
000003 01Graham Prestcott    ..............................................
000004 01Bill Somers         ..............................................
000005 01Ted Dexter          .....-........................................
000006 01Roddy Armstrong     ..............................................
000007 01Cliff Roberts       .R............................................
000008 01James Browne        .)............................................
000009 01Silvia Carrot       ...../ .......................................
000010 01Dan Peters          ."............................................
000011 01John Laws           .;............................................
000012 01Liz Childs          .?............................................
000013 01Bill McCork         .N............................................
000014 01Keith Sampson       ..............................................
000015 01John Neptune        ..............................................
Command ===> _____________________________________________________Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 23. Viewing structured data without a template

The first twenty records in this member contain non-displayable characters represented by periods and other characters. Without a template, these records cannot be easily viewed or edited.

6. When you have finished examining the data, press the Exit function key (F3) to return to the Edit (or View) Entry panel.

   Now let's view the same data using a template generated from the COBOL copybook supplied in FMN.SFMNSAM1(FMNCCPY).

7. In the **Data set name** field for the Copybook or Template, type `'FMN.V11R1M0.SFMNSAM1'`.
8. In the **Member** field for the Copybook or Template, type `FMNCCPY`.
9. In the **Copybook/template usage** field, select option **1. Above**.
10. Press Enter.

    For this example, we want to display the data in tabular format. If the **Format** field (at the top right of the panel) does not contain `TABL`, then overtype the first letter of its current value with `T` and then press Enter.

    The sample data is now shown in TABL display format, arranged in columns.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                  Top of 40
                                                  Record AT TOP    Format TABL
       REC-TYPE REC-ID NAME                 EMPLOYEE-NO     AGE   SALARY
       #2       #3 R # #4                           #5      #6       #7
       AN 1:2   AN 1:2 AN 3:20                 BI 23:2 BI 25:2  PD 27:4
       <>       <>     <---+----1----+---->      <---+>  <---+> <---+-->
****** ****  Top of data  ****
000001 01       01     Grant Smith                7712      35    75000
000002 01       01     Andrew Apple               6645      53    78500
000003 01       01     Graham Prestcott           5583      28    48000
000004 01       01     Bill Somers                4418      33    68000
000005 01       01     Ted Dexter                 3327      52    60250
000006 01       01     Roddy Armstrong            5683      34    77000
000007 01       01     Cliff Roberts              2265      57   100000
000008 01       01     James Browne               1117      46   125000
000009 01       01     Silvia Carrot              2308      29    61400
000010 01       01     Dan Peters                 4479      54    63000
000011 01       01     John Laws                  3422      21    46750
000012 01       01     Liz Childs                 3439      55    66000
Command ===>                                                   Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 24. Viewing data using a template (TABL display format)

File Manager has used the first record type defined in the copybook to display the data and the first set of records (REC-TYPE = 01) are now readable. Any fields defined in the second record type (REC-TYPE = 02) that match fields defined in the first type are also displayed.

Removing or changing the template

To stop using a template, and remove the logical view of the data, enter `TVIEW OFF` on the Command line.

To edit a template, and change the logical view of the data, enter `TEDIT` on the Command line.

To select or create a template for the data, enter `TVIEW` on the Command line.

Note: There are restrictions on removing or changing templates. When using record sampling, or when a segmented template has been used, File Manager cannot perform the changes in reformatting required. In these cases, it is necessary to exit and re-enter the editor session to perform such a change.

Related topics

Chapter 4, “Creating and editing templates,” on page 127
“TEDIT primary command” on page 824
“TVIEW primary command” on page 825

Viewing and changing packed data

When you want to view PDS or PDSE data set members that have been stored in ISPF PACK format, you can tell File Manager to unpack the data before displaying it. Once your data has been unpacked for display, you can view it as normal data. The records are automatically packed again when you save the data set.

Working with packed data in Edit

In Edit, you can convert packed data to an unpacked format and non-packed data to packed format, by issuing the PACK ON or PACK OFF command before you save your data.

To convert data to ISPF PACK format:

1. Open a data set member that is not in ISPF PACK format.
2. Perform any editing changes you want to make.
3. Enter the PACK ON command on the Command line.
4. Enter the SAVE, FILE or EXIT command.

Note: You do not need to have the **Recognize and interpret ISPF packed data** option selected in order to convert data to packed format. However, you need to have the option on, to be able to edit this data after it has been packed.

To unpack data that has been stored in ISPF PACK format:

1. Ensure that the **Recognize and interpret ISPF packed data** option has been selected in the relevant Editor Options panel.
2. Open the packed data set member.
3. Enter the PACK OFF command on the Command line.
4. Enter the SAVE, FILE or EXIT command.

Note: You cannot unpack data unless File Manager is able to recognize and interpret the packed data format.

You can also pack or unpack data set members as you copy a member to another data set or data set member.

Working with packed data in View

In View, if your PDS or PDSE data sets have been written using the ISPF PACK option, you can choose to unpack the data before viewing it, or work with the data in its packed format. This can be particularly useful when your data appears to be in ISPF PACK format, but is in fact not so.

To unpack files automatically on viewing:

1. Do one of the following:
   a. On the Primary Option Menu panel, select option **0 Settings**.
   b. On the Set Processing Options panel, select option **6 EDIT**.

   OR

   a. On the View Entry panel, select **Options** from the Action Bar.
   b. Select option 9 from the pull down menu.
2. Select the **Recognize and interpret ISPF packed data** option.

   When this option is selected, File Manager checks the data in the data set or member being viewed to determine if it has been written with the ISPF PACK option. If it had, File Manager then unpacks the data to allow it to be viewed in the normal fashion. If the unpacked data set is too large to be contained in memory, File Manager issues a message and then opens the data set in its packed format. If this option is deselected, File Manager does not check the data format and operates on packed data in its packed state.

Note: If you select the **Use I/O exit** option on the View Entry panel, the **Recognize and interpret ISPF packed data** option is ignored. Packed data sets are displayed in packed format.

Related topics

"PACK primary command" on page 796
"Viewing and changing packed data" on page 56

"Editor Options panel" on page 546
"Copying data sets" on page 253

Listing statistics for the current editor session

To list statistics for the current editor session, use the RECSTATS primary command.

The RECSTATS primary command shows (in a pop-up panel):
- The number of physical records read while loading
- The number of physical records selected
- The number of record elements in the current editor session
- For an Edit session, the number of pending inserts and changes
- The total data bytes
- The average record element length
- The minimum record element length encountered
- The maximum record element length encountered

When a template has been used, the RECSTATS primary command also shows:
- The number of layouts
- The number of record elements selected
- The number of record elements not selected
- The number of record elements not selected due to the record data not meeting selection criteria
- The number of record elements not selected due to the record length being outside any layout length
- A list of layout names and the number of record elements belonging to each layout

Note: Information on record elements not selected is not available if the **Include only selected records** option is selected and this is an in-storage or sampling editor session.

Similarly, for an editor session using a segmented template with selection criteria, information for records and their segments that were not selected is not available.

Figure 25 on page 59 shows an example of statistics listed by the RECSTATS command for a editor session when a template has been used.

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                 Rec 0 of 40
                                  Record 0         Col 1           Format CHAR
┌──────────────────────────── File Manager Messages ─────────────────────────┐
│                                                                            │
│ File Manager View (in-storage)                                             │
│ Dataset: FMN.HOGN.KSDS.TEST                                                │
│ Template: FMN.RFM0569.PDSE(HOGAN)                                          │
│ Physical records read while loading: 100                                   │
│ Physical records selected          : 2                                     │
│                                                                            │
│ Number of record elements    :  50                                        │
│   inserts pending            :  0                                         │
│   changes pending            :  0                                         │
│   deletes pending            :  0                                         │
│ Total data bytes             :  4014                                       │
│ Average record element length:  80                                         │
│ Minimum record element length:  12                                         │
│ Maximum record element length:  298                                        │
│                                                                            │
│ Template related information                                               │
│ Number of layouts            :  28                                         │
│ Record elements selected     :  50                                         │
│ Record elements not selected :  0                                          │
│   due to record element data not meeting selection criteria:  0            │
│   due to record element length outside any layout length   :  0            │
│ List of layout names and count of record elements per layout               │
│ M-KEY-GROUP                                                              2 │
│ M-ACCOUNT-CODING-STRUCTURE                                               2 │
│ M-ORIGINAL-ACT-INFO                                                      2 │
│ M-CLOSED-ACT-INFO                                                        0 │
│ Command ===> ____________________________________________ Scroll PAGE      │
│  F1=Help       F2=Split      F3=Exit       F4=CRetriev   F7=Backward     │
│  F8=Forward    F9=Swap      F10=Actions    F12=Cancel                     │
└────────────────────────────────────────────────────────────────────────────┘
```

Figure 25. Example of statistics listed with the RECSTATS command

Related topics

"RECSTATS primary command" on page 804

Displaying your current editor options

To display your current editor options while in an editor session (from the Edit panel or View panel), enter the PROFILE primary command.

Figure 26 on page 60 shows an example of editor options displayed during an editor session.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMNUSER.DATA(DATA1)                            Rec 0 of 40
                                                  Record 0         Format TABL
REC-TYPE NAME                  EMPLOYEE-NO     AGE   SALARY    MONTH(1)
#2       #3                          #4       #5       #6          #7
AN 1:2   AN 3:20                 BI 23:2 BI 25:2  PD 27:4     BI 31:4
<>       <---+----1----+---->      <---+>  <---+> <---+--> <---+----1>
****  Top of data  ****
....BOUNDS 1-*...CAPS OFF...CASE OFF...PACK OFF...PREFIX OFF...................
....SHADOW LINES....EXCLUDED ON..NOTSEL ON..SUPPRESSED ON......................
....SHOW(UNGROUP)....NOTSEL OFF...SUPPRESSED OFF...............................
....TEMPLATE FMNUSER.DATA.......................................................
....JUST OFF...OFFSET 0...PIC OFF...RDF ON...REFS ON...........................
....SLOC OFF...STR OFF...TYPE ON................................................
01       Grant Smith                7712      35    70000           6
01       Andrew Apple               6645      53    78500          30
01       Graham Prestcott           5583      28    48000           7
01       Bill Somers                4418      33    68000           5
01       Ted Dexter                 3327      52    60250          14
01       Roddy Armstrong            5683      34    77000          28
Command ===>                                                    Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 26. Editor options displayed during an editor session

Related topics
"PROFILE primary command" on page 800

Ending an editor session

In View, to end your editor session, press the Exit function key (F3).

Note: If you have selected more than one data set member to be viewed, the next selected member on the list is now displayed.

In Edit, to end your editor session without saving your changes (that is, since the previous SAVE command), enter CANCEL (or CAN) on the Command line, or press the Cancel function key (F12). If you have made any changes to the data, File Manager displays a message box asking you to confirm that you want to discard the changes.

To end your Edit session and save any changes you have made, enter END on the Command line, or press the Exit function key (F3).

Related topics
"END primary command" on page 762
"FILE primary command" on page 769
"CANCEL primary command" on page 744

Saving changes without ending the Edit session

To save changes you have made to the current data set or member, without ending the Edit session, use the SAVE command.

File Manager keeps only changed records in memory, unless using a MEMORY record limit, or using record sampling, or using a segmented template with selection criteria. If you run out of memory, the SAVE command may free some memory, which would let you continue the Edit session.

Note: This command is not supported when editing a large file using an auxiliary data set. In this case, you must end the Edit session to save the changes and then re-edit the data set if you want to make further changes.

Related topics
"SAVE primary command" on page 812

Displaying the RBA and record length information

When you are viewing a VSAM file in a Browse editor session, you can select to display the RBA and record length information.

The display of this information is initially controlled by the **Display RBA and Length when browsing VSAM** option in the relevant Editor Options panel. However, in a Browse session you can use the RBALEN primary command to change whether this information is displayed or not for that editor session.

In a Browse session, to display the RBA and record length when viewing a VSAM file, do one of these:

- Ensure that the **Display RBA and Length when browsing VSAM** option in the relevant Editor Options panel is selected, or
- Issue the command `RBALEN ON`

Figure 27 shows part of a VSAM file displayed in a Browse session with the RBA and record length displayed on the leftmost side.

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
Browse           FMN.RFM0201.KSDS.REUSE.SHR1                        Rec 0
       Type KSDS     Key ____________________     RBA __________ Format CHAR
       Col 1________
RBA         Len    -<==+===>=----+----2----+----3----+----4----+----5----+----6
****  Top of data  ****
0             80    00000001AAA*** updateok & reversed ***AAAAAAAAAAAAAAAAAAAA
80            80    00000002AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
160           80    00000003AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
240           80    00000004AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
320           80    00000005AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
400           80    00000006AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
480           80    00000007AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
560           80    00000008AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
640           80    00000009AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
720           80    00000010AAA*** updateok & reversed ***AAAAAAAAAAAAAAAAAAAA
800           80    00000011AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
880           80    00000012AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
960           80    00000013AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
1040          80    00000014AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Command ===> ____________________________________________________ Scroll CSR
 F1=Help    F2=Zoom    F3=Exit    F4=Expand  F5=RFind   F7=Up      F8=Down
 F9=Swap   F10=Left   F11=Right  F12=Cancel
```

Figure 27. VSAM file displayed in Browse session with RBA and record length details displayed

To display the VSAM file without the RBA and record length details, perform either of these actions:

- Ensure that the **Display RBA and Length when browsing VSAM** option in the relevant Editor Options panel is not selected,

- Issue the command RBALEN OFF

Figure 28 shows the same VSAM file displayed in a Browse session with the RBA and record length not shown.

```
 Process   Options   Help
------------------------------------------------------------------------------
Browse           FMN.RFM0201.KSDS.REUSE.SHR1                          At top
       Type KSDS     Key ____________________     RBA __________ Format CHAR
       Col 1________
-<==+===>=----+----2----+----3----+----4----+----5----+----6----+----7----+-
****  Top of data  ****
00000001AAA*** updateok & reversed ***AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000002AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000003AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000004AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000005AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000006AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000007AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000008AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000009AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000010AAA*** updateok & reversed ***AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000011AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000012AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000013AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000014AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Command ===> ____________________________________________________ Scroll CSR
 F1=Help    F2=Zoom    F3=Exit    F4=Expand  F5=RFind   F7=Up      F8=Down
 F9=Swap   F10=Left   F11=Right  F12=Cancel
```

Figure 28. VSAM file displayed in editor session with RBA and record length details not displayed

Related topics

"RBALEN primary command" on page 800
"Editor Options panel" on page 546

Managing the prefix area

The prefix area displays line numbers that represent the record number of each line in the data set being edited.

The prefix area also doubles as the prefix command entry area. You enter prefix commands by overtyping the line number for the appropriate record.

You can use prefix commands to:

- Insert or delete lines
- Repeat (duplicate) lines
- Move lines
- Shift data
- Limit the data being edited

The prefix area also serves to:

- Highlight unidentified segments when you are working with segmented records with related ID criteria.
- Identify records not selected because of invalid length when you are working with variable-length arrays.
- Indicate why a record is not selected in an editor session.

Related topics
"How File Manager handles segmented data with related ID criteria" on page 176
"Support for variable-length arrays" on page 136
"Seeing why a record is not-selected" on page 227

In an editor session, to see which prefix commands are valid for the current environment, enter an "?" in the prefix area (against any listed line). File Manager displays a list of the line commands you can enter, like those shown in Figure 29.

```
 Process   Options   Utilities   Help
                File Manager Available Edit Prefix Commands
                                                                More:     +

You can enter the following commands in the prefix area of the data display:

A     identifies the record after which data is to be moved or copied.

B     identifies the record before which data is to be moved or copied.

BND   place a bounds record here to show current editor bounds.

C     copy one record.
Cn    copy n records.
CC    copy block of records. Mark start and end of block.

COLS  Use COLS to display the columns line.  For example:

         000300
         cols   ----+----10---+----2----+----3----+----4----+----5---    etc.
         000500

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 29. Using the ? prefix command to display a list of valid prefix commands

In an editor session, you can choose whether a prefix area is displayed next to the data or not.

To specify the default position and display state of the prefix area for *all* editor sessions, set these editor options on the relevant Editor Options panel:
- **Display prefix area**
- **Display prefix area on the right**
- **Prefix area width (A,6,7,8,9)**

To specify the default position and display state of the prefix area for the *current* editor session, use the PREFIX primary command. For example:
- To display the prefix area on the right, enter `PREFIX RIGHT`.
- To display the prefix area as an 8-digit field, enter `PREFIX 8`.
- To display the prefix area with a width that automatically adjusts (from 6 to 9 digits wide) in order to display the record number, enter `PREFIX A`.
- To remove the display of the prefix area, enter `PREFIX OFF`.

You can use most prefix commands on either a single line or a block of consecutive lines:
- To perform an operation on a single line, you enter the appropriate prefix command.

- To perform an operation on a block of consecutive lines, either enter the prefix command preceded or followed by the number of lines, or enter the appropriate block prefix command at the start and end lines of the block.

Generally, you need to type over only the first 1 or 2 characters of the line number to enter a prefix command. Sometimes, however, typing a single character can be ambiguous. In the following example, it is unclear whether the intended prefix command is R to repeat line 31700, or R3 to repeat the line three times:

```
031600
R31700
031800
```

In such cases, the editor assumes that you have not typed a number following the prefix command. If you want to repeat the line three times, you can use any of the following methods:

- Type one or more blanks following the R3:

  ```
  R3 700
  ```

- Type R3 and press the Erase EOF key to clear the rest of the Line Command field, or press the Erase EOF key and then type R3.
- Type one or more blanks after the R but before the number such that the number when entered is different than the characters being overtyped.:

  ```
  R 3700
  ```

- Type the number before the R, ensuring that the number when entered is different than the characters being overtyped:

  ```
  3R1700
  ```

To clear any outstanding commands in the prefix area, use the RESET PREFIX command. See “RESET primary command” on page 806.

Prefix commands do not affect not-selected or suppressed records that are hidden from display or represented by shadow lines. For example, if you use the MM prefix command to mark a block of records for moving, then not-selected or suppressed records that are hidden from display or represented by shadow lines are not affected by the move, and are left behind at their original position. Prefix commands **do** affect excluded records, even when they are hidden from display (by SHOW EX OFF and SHADOW EX OFF).

Related topics

“PREFIX primary command” on page 798

Manipulating your view of the data

Once your data has been displayed in the editor panel, you can manipulate your view of the data to focus on the records of interest to you.

The tasks described in this section are:

- “Selecting a display format” on page 66;
- “Zooming in to see all of a record” on page 71
- “Scrolling to see data” on page 77;
- “Sorting the data” on page 82; and
- “Finding specific data” on page 84.

Using primary commands in an editor session

When you are viewing or editing data, you can enter primary commands on the Command line of the editor panel to perform the following functions:

Action Command

Scroll the data
- "BOTTOM primary command" on page 742
- "DOWN primary command" on page 760
- "LEFT primary command" on page 784
- "NEXT primary command" on page 793
- "PREVIOUS primary command" on page 798
- "RIGHT primary command" on page 810
- "TOP primary command" on page 824
- "UP primary command" on page 826

Scroll to a particular record number or text
- "LOCATE primary command" on page 786

Find a string
- "FIND/FX primary command" on page 769
- "RFIND primary command" on page 809

Change a string
- "CHANGE/CX primary command" on page 748
- "RCHANGE primary command" on page 801

Switch between display formats
- "FORMAT primary command" on page 780
- "FORMAT primary command" on page 780
- "FORMAT primary command" on page 780
- "FORMAT primary command" on page 780
- "FORMAT primary command" on page 780

Turn off highlighting of found strings
- "RESET primary command" on page 806

Find fields in error
- "FE (Find Error) primary command" on page 767

Change data in error
- "CE (Change Error) primary command" on page 746

Print a record
- "RD primary command" on page 801
- "RP primary command" on page 811

Obtain record and selection statistics
- "RECSTATS primary command" on page 804

Hide or show shadow lines
- "SHADOW primary command" on page 816

Group or ungroup record sets
- "SHOW primary command" on page 817

Zoom in to see all of one record
- "ZOOM primary command" on page 829

View suppressed records
- "VIEW primary command" on page 828

Display data in hexadecimal format
"HEX primary command" on page 780

Show or hide the Field Reference and Redefines columns when in SNGL display
"REFS primary command" on page 805

Show or hide the Field Type and Length columns when in SNGL display
"TYPE primary command" on page 826

Remove or change the logical view
"TEDIT primary command" on page 824
"TVIEW primary command" on page 825

Adjust the order of lines
"SORT primary command" on page 818

Display the RBA and record length information
"RBALEN primary command" on page 800

Related topics
Chapter 15, "Primary commands," on page 739

Selecting a display format

A display format is a way of arranging your data within the data area of the editor panel, to suit your viewing or editing needs. Display formats can be applied to data that has or has not been given a logical format via a template. Data that has not been formatted with a template can be shown in CHAR, HEX or LHEX display format. Data that has been formatted with a template, can be shown in these formats, as well as in SNGL or TABL display format.

Setting the initial display format

To specify how the editor panel appears at the start of an editor session, set the **Initial Display** option on the relevant Editor Options panel to the required display format.

Changing the display format

To change the display format:

1. Position your cursor in the **Format** field (upper right corner).
2. Overtype the current setting with the first letter of the required format and press Enter. The panel redisplays in the selected format.

   OR
3. Enter the FC (FORMAT CHAR), FH (FORMAT HEX), FL (FORMAT LHEX), FS (FORMAT SNGL) or FT (FORMAT TABL) primary commands.

Note: The abbreviated form of these commands can also be entered in any prefix command field.

Display formats

To determine which display format is best for your needs, review the display format descriptions below.

CHAR
Character display format. This format is suitable for viewing or editing unstructured text that is composed of characters that can be entered via the keyboard.

```
                                Record 0        Col 1          Format CHAR
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
****  Top of data  ****
       01 REC-TYPE01.                                                0008000
          03 REC-TYPE             PIC XX.                           0009000
          03 NAME                 PIC X(20).                        0010000
```

HEX Hexadecimal updown format, followed by a scale. This format uses separate lines; one for the characters, one for the zone digit and one for the numeric digit of each byte. Hexadecimal updown format is suitable for editing data that is composed of characters or values that cannot be entered via a keyboard or cannot be displayed. For example, you may want to use the HEX display format when editing DBCS characters or packed decimal numbers.

```
                                Record 0        Col 1          Format HEX

****  Top of data  ****
       01 REC-TYPE01.                                                0008000
4444444FF4DCC6EEDCFF44444444444444444444444444444444444444444444444444FFFFFFF
000000001095303875O1B0000000000000000000000000000000000000000000000000008000
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----

          03 REC-TYPE             PIC XX.                           0009000
4444444444FF4DCC6EEDC4444444444444DCC4EE44444444444444444444444444444FFFFFFF
000000000003095303875000000000000793077B0000000000000000000000000009000
```

LHEX Unformatted hexadecimal. This format has been superceded by the HEX format. However, it has been retained within the product so that long-term users of File Manager can continue working with Hexadecimal characters in a familiar layout.

```
                               Record 0        Col 1          Format LHEX
- - - - + - - - - 1 - - - - + - - - - 2 - - - - + - - - - 3 - - - - + - - - -
****  Top of data  ****
404040404040F0F140D9C5C360E3E8D7C5F0F14B4040404040404040404040404040404040
4040404040404040404040F0F340D9C5C360E3E8D7C5404040404040404040404040404040D7C9C340
4040404040404040404040F0F340D5C1D4C5404040404040404040404040404040404040D7C9C340
4040404040404040404040F0F340C5D4D7D3D6E8C5C560D5D6404040404040404040404040D7C9C340
```

When in Zoom mode, the selected record displays as formatted hexadecimal and characters (system dump format). For example:

```
                                                    Col 1          Format LHEX
Record Length Byte ---------------- Hex -----------------   ------ Char ------
1          80 0000 40404040  404040F0  F140D9C5  C360E3E8   *       01 REC-TY*
              0010 D7C5F0F1  4B404040  40404040  40404040   *PE01.           *
              0020 40404040  40404040  40404040  40404040   *                *
              0030 40404040  40404040  40404040  40404040   *                *
              0040 40404040  40404040  F0F0F0F8  F0F0F0F0   *        00080000*
```

SNGL Single-record format (only available when using a template). Limits the display to a single record. Each field might take up one or more lines, depending upon the field information selected for display and the length of the field name. SNGL display format is suitable for viewing information that has been structured into fields and records, when you need to focus on a single record.

```
View            FMN.V11R1M0.SFMNSAM1(FMNCDATA)                Rec 1 of 40
                                                                 Format SNGL
                                 Top Line is 1    of 11    in Record 17
Ref Field    Picture Typ Start   Len   Data
  1 1 REC-TYPE02
                     AN      1    80
  2  2 REC-TYPE
            XX       AN      1     2   02
  3  2 NAME  X(20)    AN      3    20   Bob McDonald
  4  2 JOB-TITLE
            X(14)    AN     23    14   Programmer
  5  2 ADDR1 X(20)    AN     37    20   67  Hampshire Road
  6  2 ADDR2 X(20)    AN     57    20   Parkwood
  7  2 POSTCODE
            X(4)     AN     77     4   4507
****  End of record  ****
```

This example shows the Ref (Field Reference), Field, Picture, Typ (Type), Start and Len (Length) columns, and the Field column contains structure information (copybook element level). The display of many of these columns is optional (the default is for them to be displayed).

To change the information displayed in SNGL mode, enter one of the following commands:

JUST Left justifies the numeric fields

PIC Displays or hides the Picture column

RDF Displays or hides the Redefines information (within the Field column) and the redefined fields. This command also affects the display of the redefined fields in TABL display.

REFS Displays or hides the Field References column

SLOC Displays or hides the Start column

STR Displays or hides the structure information in the Field column

TYP Displays or hides the Type and Length columns

CCSID
Displays or hides the CCSID column.

These are toggle commands that change the display from on to off or off to on, depending on the current status of the panel. The commands only affect the current editor session, they do not change the default settings.

To change the default settings for editor panel:

1. Ensure that you are not currently in an editor session.
2. Choose **Options**> **Editor options** from the Primary Options menu.
3. Select or clear the following record formatting options for SNGL display or print in the relevant Editor Options panel:
   - Field reference number
   - Field type and length values
   - Picture clause
   - Start location
   - Structure
   - Left justify numeric fields
   - Redefined fields

Note: These options cannot be changed when you are in an editor session.

TABL Tabular format (only available when using a template). Arranges fields in

columns. This format is suitable for viewing information that has been structured into fields and records and you need to review multiple records.

```
REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY   MONTH(1)
#2       #3                         #4       #5       #6         #7
AN 1:2   AN 3:20               BI 23:2 BI 25:2  PD 27:4    BI 31:4
<>       <---+----1----+---->     <---+-> <---+-> <---+--> <---+---->
01       Grant Smith                7712      35    75000          6
01       Andrew Apple               6645      53    78500         30
:
```

In TABL display format, only one record type in the data set is displayed at any time. The column headings at the top of the screen reflect the fields in the current record type. There are three lines in the headings:

```
field_heading
#n [R #n]
type start_column:length
<- ... ->
```

where:

field_heading
: Defaults to the field name defined in the copybook. You can change this by editing the template. For details, see Chapter 4, “Creating and editing templates,” on page 127.

#*n* [R #*n*]
: The field reference. You use this to refer to the field in File Manager. For example, to limit the FIND primary command to searching only certain fields, you specify the field references for those fields.

  If the field redefines another field, then this is shown by the “R” notation, followed by the field it redefines. For example, `#5 [R #4]` shows that this is field 5, which redefines field 4.

type
: The field data type.

start_column
: The field start column.

length
: The field length.

<- ... ->
: Ruler, indicating the size of the field.

Records of other types are “suppressed” from display: they are either hidden or represented by shadow lines.

In CHAR, HEX and LHEX formats, and for alphanumeric fields in SNGL or TABL formats, characters that cannot be displayed (“non-displayable” characters) are represented as periods.

- You can display hexadecimal values under your formatted field data by entering `HEX ON` on the Command line. For example, if you are viewing data in TABL display format and your display looks something like this:

  ```
  Order# # Items Date   Time   Seller  Buyer     (1)    (1)
  #6          #7 #9     #13       #18    #19     #31    #32
  A00001       1 900227 010101   1144     20      25   9999
  A00004       1 900310 100530   1144     10      15   0099
  ```

  then issuing the command HEX ON causes the hexadecimal representation to be displayed:

```
Order# # Items Date   Time   Seller  Buyer     (1)    (1)
#6          #7 #9     #13      #18    #19     #31    #32
A00001       1 900227 010101   1144     20      25   9999
CFFFFF   00001 FFFFFF FFFFFF   0014   0000    0001   0020
100001   0000F 900227 010101   014F   002F    0009   007F

A00004       1 900310 100530   1144     10      15   0099
CFFFFF   00001 FFFFFF FFFFFF   0014   0000    0000   0006
100004   0000F 900310 100530   014F   001F    000F   0003
```

- If a numeric field contains data that cannot be correctly interpreted as a numeric value (for example, a packed decimal field contains invalid packed values), then the field is displayed as highlighted asterisks.
- If a numeric field contains a valid numeric value, but the value is too large to fit in the width allocated to the field on the screen (as defined in the picture specification of the original copybook), then the value is truncated. The field is highlighted to indicate that truncation has occurred.

  To show the entire field without truncation, you need to edit the template, and specify an output width for the field.
- If a field is a array element, then its field heading is followed by a subscript in parentheses; for example, ELEMENT(1).

Related topics

"FORMAT primary command" on page 780
"HEX primary command" on page 780
"Zooming in to see all of a record" on page 71
"REFS primary command" on page 805
"TYPE primary command" on page 826
"Editor options (option 0.6)" on page 45
"Editor Options panel" on page 546
"Filtering record display using templates" on page 224

Displaying a column identification line

In an editor session, when you are in CHAR, HEX, or LHEX display format, you can enter the COLS prefix command to display a column identification line. The column identification line is identical to the line already shown at the top of the data but can be useful in checking the position of data in a record.

To display the column identification (=COLS>) line:

1. Type COLS in the prefix area of any line.
2. Press Enter.

   The column identification line is inserted in the data set or member after the line in which you entered COLS. The column identification line moves with the rest of the data when you scroll through the data set or member.

Note: You can use the COLS prefix command with the BOUNDS prefix command to help check and reposition the bounds settings.

To remove the column identification line from the panel, either type D in the prefix area that contains the =COLS> flag, or type RESET on the command line:

Figure 30 on page 71 shows the boundary definition line with the COLS command typed in the prefix area of the following record.

```
 Process   Options   Help
------------------------------------------------------------------------------
Edit            JOHNLEV.TEST.DATA(EMP)                          Top of 40
       Col 1      Insert Length 80          Record AT TOP     Format LHEX
        - - - - + - - - - 10 - - - + - - - - 2 - - - - + - - - - 3 - - - - + -
****** ****  Top of data  ****
000001 F0F1C7998195A340E2A4A38885999381958440404040401E2000380075111F000000060000
000002 F0F1C195849985A640C1A2A39385404040404040404019D400150075000F0000001E0000
000003 F0F1C7998188819440D7A4998489854040404040404015880018006800 0F000000070000
000004 F0F1C289939340E2969785994040404040404040404011AB001C0068000F000000050000
000005 F0F1E3A89996958540C481938189A2404040404040400C6200150060000F0000000E0000
BND        <                                  >
cols06 F0F1D9968440E3A4999585994040404040404040404138E00230100000F0000001C0000
000007 F0F1C39389A58540D5858193969540404040404040404008D1002F0100000F0000002C0000
000008 F0F1D1899440C19385A781958485994040404040404040534003F0125000F0000002F0000
000009 F0F1E28993A581959640D79985A9404040404040404040B4F00320060000F000000010000
000010 F0F1C4969540D7888199968188404040404040404040131A00370063000F000000260000
000011 F0F1D196889540D385A599899587A3969540404040400D1000280067000F0000000E0000
000012 F0F1D389A940D9A4A288A3969540404040404040404040ECD00280066000F0000002C0000
000013 F0F1C2968240D483C396999481839240404040404040412FD00280076000F0000000A0000
000014 F0F1D28589A38840E2A385A68199A34040404040404040084100280070000F0000001A0000
Command ===> ____________________________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 30. Before the COLS prefix command

When you press Enter, File Manager inserts the COLS line, as shown in Figure 31.

```
 Process   Options   Help
------------------------------------------------------------------------------
Edit            JOHNLEV.TEST.DATA(EMP)                          Top of 40
       Col 1      Insert Length 80          Record AT TOP     Format LHEX
        - - - - + - - - - 10 - - - + - - - - 2 - - - - + - - - - 3 - - - - + -
****** ****  Top of data  ****
000001 F0F1C7998195A340E2A4A38885999381958440404040401E2000380075111F000000060000
000002 F0F1C195849985A640C1A2A39385404040404040404019D400150075000F0000001E0000
000003 F0F1C7998188819440D7A4998489854040404040404015880018006800 0F000000070000
000004 F0F1C289939340E2969785994040404040404040404011AB001C0068000F000000050000
000005 F0F1E3A89996958540C481938189A2404040404040400C6200150060000F0000000E0000
BND        <                                  >
=COLS>  - - - - + - - - - 10 - - - + - - - - 2 - - - - + - - - - 3 - - - - + -
000006 F0F1D9968440E3A4999585994040404040404040404138E00230100000F0000001C0000
000007 F0F1C39389A58540D5858193969540404040404040404008D1002F0100000F0000002C0000
000008 F0F1D1899440C19385A781958485994040404040404040534003F0125000F0000002F0000
000009 F0F1E28993A581959640D79985A9404040404040404040B4F00320060000F000000010000
000010 F0F1C4969540D7888199968188404040404040404040131A00370063000F000000260000
000011 F0F1D196889540D385A599899587A3969540404040400D1000280067000F0000000E0000
000012 F0F1D389A940D9A4A288A3969540404040404040404040ECD00280066000F0000002C0000
000013 F0F1C2968240D483C396999481839240404040404040412FD00280076000F0000000A0000
Command ===> ____________________________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 31. After the COLS prefix command

Zooming in to see all of a record

In CHAR, HEX, or LHEX display format, if you want to show all of the data in a particular record without having to scroll left or right, you can "zoom in" by moving the cursor to the record, then pressing the Zoom function key (F2). File Manager limits the display to just that record, as shown in Figure 32 on page 72.

The zoomed display automatically scrolls to the column that was shown in the **Col** field prior to zooming.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                Rec 1 of 40
                                                        Col 1_________ Format CHAR
Record Length Byte ----+----1----+----2----+----3----+----4----+----5----+----6
40         80    1 02Grant Smith         Developer     22  Montrose St     Thor
                61 nlie            6145

Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 32. Zooming in on one record (CHAR display format)

Note: This example shows data from the second record type in the sample data set, as that record type holds more character information.

To zoom out, press the Zoom function key (F2) again. Zooming toggles between zooming in on one record and zooming out to show multiple records. When you zoom out, the display format returns to its setting prior to zooming in. For example, if you zoom in when the display format is CHAR, then change to SNGL, zooming out returns the display to CHAR format.

In SNGL or TABL display formats, when you zoom in on a record, just that record is displayed in "zoomed" SNGL display format, as shown in Figure 33 on page 73. This format differs from "normal" SNGL display format, in that:

- In this display format, you cannot navigate between records.

  In "un-zoomed" SNGL display format, you can navigate between records using the PREVIOUS and NEXT primary commands or Previous function key (F10) and Next function key (F11).
- When you view a record in "zoomed" SNGL display format, File Manager displays all fields, regardless of whether or not they were selected for display in the template.

```
 Process   Options   Help
--------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                 Rec 1 of 40
Current type is REC-TYPE02                                     Zoom Format SNGL
                                  Top Line is 1    of 11    in Record 21
Ref Field    Picture Typ Start   Len   Data
  1 1 REC-TYPE02
                     AN      1    80
  2  2 REC-TYPE
            XX       AN      1     2   02
  3  2 NAME  X(20)    AN      3    20   Grant Smith
  4  2 JOB-TITLE
            X(14)    AN     23    14   Developer
  5  2 ADDR1 X(20)    AN     37    20   22  Montrose St
  6  2 ADDR2 X(20)    AN     57    20   Thornlie
  7  2 POSTCODE
            X(4)     AN     77     4   6145
****  End of record  ****









Command ===>                                                    Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 33. Zooming in on one record ("zoomed" SNGL display format)

Note: SNGL view with the STR command active provides a view of your record structure that is identical to the ZOOM SNGL view and allows you to scroll between the records.

In TABL or normal ("un-zoomed") SNGL display format, File Manager only shows the fields selected for display in the template. For instance, in the example shown in Figure 33, if you edit the template to deselect the field JOB-TITLE, in "un-zoomed" SNGL display format the record looks like this:

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)               Rec 21 of 40
Current type is REC-TYPE02                                        Format SNGL
                                  Top Line is 1    of 7     in Record 21
Ref Field    Picture Typ Start   Len   Data
  2 REC-TYPE
              XX      AN      1     2   02
  3 NAME      X(20)   AN      3    20   Grant Smith
  5 ADDR1     X(20)   AN     37    20   22  Montrose St
  6 ADDR2     X(20)   AN     57    20   Thornlie
  7 POSTCODE
              X(4)    AN     77     4   6145
****  End of record  ****
```

Figure 34. SNGL format with unselected fields

In TABL display format, zooming in automatically scrolls to the field that was at the left of the screen when the display format was TABL.

Related topics

"ZOOM primary command" on page 829
"STR primary command" on page 821
"Selecting a display format" on page 66
"View panel" on page 707
"Editor panel" on page 534
"Changing the length of a record" on page 125

Displaying the record length

When you are viewing a file in a multi-line display format (TABL, CHAR, HEX, or LHEX), you can select to display the record length of each record.

Note: In SNGL and zoomed display formats, the record length is *always* displayed.

The display of this information is initially controlled by options in the Editor Options panels (which apply to each new editor session):

Display record length
: Turns on the display of record lengths. By default, File Manager displays the record length on the left of the data.

Display record length on the right
: Displays the record length on the right of the data (only when the **Display record length** option is selected or if you issue the `RECLEN ON` command during an editor session).

Record length width (A,3,4,5,6)
: Controls the width of the record length display area.

However, when viewing data in an editor session you can use the RECLEN primary command to temporarily change the record length display options for the current editor session:

RECLEN ON
Turn on the display of record lengths. (Record lengths are displayed on the left of the data, unless the **Display record length on the right** editor option is selected.)

RECLEN OFF
Turn off the display of record lengths.

RECLEN RIGHT
Turn on the display of record lengths and display record lengths on the right of the data.

RECLEN *n*
Set the width of the record length display area to *n* characters, where *n* can be 3, 4, 5, or 6.

RECLEN A
Set the width of the record length display area according to the value of the record lengths.

Figure 27 on page 61 shows part of a file displayed in an editor session with no record length information displayed.

```
 Process   Options   Help
--------------------------------------------------------------------------------
View             SOPERW.JCL(FMNCDATA)                          Top of 40
                                                  Record AT TOP   Format TABL
       REC-TYPE NAME                  EMPLOYEE-NO     AGE   SALARY    MONTH(1)
       #2       #3                          #4       #5       #6          #7
       AN 1:2   AN 3:20                 BI 23:2 BI 25:2  PD 27:4     BI 31:4
       <>       <---+----1----+---->      <---+>  <---+> <---+--> <---+----1>
****** ****  Top of data  ****
000001 01       Grant Sutherland           7712      96    75111           6
000002 01       Bndrew Bstle               6612      21    75000          30
000003 01       Graham Purdie              5512      94    68000           7
000004 01       Bill Soper                 4412      28    68000           5
000005 01       Tyrone Dalais              3312      21    60000          14
000006 01       Rod Turner                 5612     100   100000          28
000007 01       Clive Nealon               2212     100   100000          44
000008 01       Jim Blexander              1112     110   125000          47
000009 01       Silvano Prez               2312      50    60000           1
000010 01       Don Pharoah                4412      55    63000          38
000011 01       John Levrington            3412      40    67000          14
000012 01       Liz Rushton                3412      40    66000          44
Command ===>                                                        Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 35. Data displayed in an editor session with no record length details shown

To display the record length information on the left of the data, enter:

```
RECLEN LEFT
```

Alternatively, if the **Display record length on the right** option on the relevant Editor Options panel is *not* selected, you can enter:

```
RECLEN ON
```

Figure 36 shows the same data displayed with the record length displayed on the left of the data.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             SOPERW.JCL(FMNCDATA)                          Top of 40
                                                    Record AT TOP    Format TABL
           REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY
           #2       #3                         #4       #5       #6
           AN 1:2   AN 3:20                BI 23:2 BI 25:2  PD 27:4
           <>       <---+----1----+---->     <---+>  <---+> <---+-->
******     ****  Top of data  ****
000001  80 01       Grant Sutherland           7712      96    75111
000002  80 01       Bndrew Bstle               6612      21    75000
000003  80 01       Graham Purdie              5512      94    68000
000004  80 01       Bill Soper                 4412      28    68000
000005  80 01       Tyrone Dalais              3312      21    60000
000006  80 01       Rod Turner                 5612     100   100000
000007  80 01       Clive Nealon               2212     100   100000
000008  80 01       Jim Blexander              1112     110   125000
000009  80 01       Silvano Prez               2312      50    60000
000010  80 01       Don Pharoah                4412      55    63000
000011  80 01       John Levrington            3412      40    67000
000012  80 01       Liz Rushton                3412      40    66000
Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 36. Record length details displayed in editor session on left of data

To display the record length information on the right of the data, enter:

```
RECLEN RIGHT
```

Alternatively, if the **Display record length on the right** option on the relevant Editor Options panel *is* selected, you can enter:

```
RECLEN ON
```

Figure 37 on page 77 shows the same data displayed with the record length displayed on the right of the data.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             SOPERW.JCL(FMNCDATA)                          Top of 40
                                                  Record AT TOP  Format TABL
       REC-TYPE NAME                  EMPLOYEE-NO     AGE   SALARY
       #2       #3                          #4       #5       #6
       AN 1:2   AN 3:20                  BI 23:2 BI 25:2  PD 27:4
       <>       <---+----1----+---->      <---+>  <---+> <---+-->
****** ****  Top of data  ****
000001 01       Grant Sutherland            7712      96    75111            80
000002 01       Bndrew Bstle                6612      21    75000            80
000003 01       Graham Purdie               5512      94    68000            80
000004 01       Bill Soper                  4412      28    68000            80
000005 01       Tyrone Dalais               3312      21    60000            80
000006 01       Rod Turner                  5612     100   100000            80
000007 01       Clive Nealon                2212     100   100000            80
000008 01       Jim Blexander               1112     110   125000            80
000009 01       Silvano Prez                2312      50    60000            80
000010 01       Don Pharoah                 4412      55    63000            80
000011 01       John Levrington             3412      40    67000            80
000012 01       Liz Rushton                 3412      40    66000            80
Command ===>                                                   Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 37. Record length details displayed in editor session on right of data

To set the width of the field length display area for all editor sessions, set the **Record length width** option on the relevant Editor Options panel to A, 3, 4, 5, or 6. Setting this option to "A" causes the width of the record length area to be automatically adjusted as required.

To change the width of the field length display area for the current editor session, issue the RECLEN command. For example, to change the width of the field length display area to 4, enter:

```
RECLEN 4
```

Related topics

"RECLEN primary command" on page 803
"Editor Options panel" on page 546
"Changing the length of a record" on page 125

Scrolling to see data

To scroll through your data, use the following function keys:

Use this function key
To do this...
Scroll backward (up)
Up function key (F7)
Scroll forward (down)
Down function key (F8)
Next function key (F11)
Go to the next record
Previous function key (F10)
Go to the previous record
Scroll right
Right function key (F11)
Scroll left
Left function key (F10)

You can also scroll using the primary commands UP, DOWN, LEFT, RIGHT, TOP, BOTTOM, NEXT and PREVIOUS. For details, see Chapter 15, "Primary commands," on page 739.

In SNGL display format, to move forward *n* records, perform either of these actions:

- Enter NEXT *n* on the Command line
- Enter *n* on the Command line and press Next function key (F11)

Note: If you specify a number of records greater than the number of subsequent records, File Manager displays the "**** End of data ****" indicator. You can then press the Previous function key (F10) to view the last selected record.

In SNGL display format, to move back *n* records, perform either of these actions:

- Enter PREVIOUS *n* on the Command line
- Enter *n* on the Command line and press Previous function key (F10)

Note: If you specify the MAX parameter, the first selected record is displayed. If you specify a number of records greater than the number of prior records, File Manager displays the "**** Top of data ****" indicator. You can then press the Next function key (F11) to view the first selected record.

Examples

LEFT 8 Scrolls 8 columns to the left.

LEFT Scrolls left the number of columns indicated by the **Scroll** field.

LEFT CSR
If the cursor is positioned on a record, scrolls left to the cursor position; otherwise scrolls left one page.

UP 15 Scrolls up 15 lines.

UP Scrolls up the number of lines indicated by the **Scroll** field.

UP DATA
Scrolls up one line less than a page of data.

Related topics
"UP primary command" on page 826
"DOWN primary command" on page 760
"LEFT primary command" on page 784
"RIGHT primary command" on page 810
"PREVIOUS primary command" on page 798
"NEXT primary command" on page 793
"TOP primary command" on page 824
"BOTTOM primary command" on page 742

Controlling how far you scroll

To control how far you scroll when you press one of the scrolling function keys, enter a scroll amount in the **Scroll** field:

Scroll amount
Scrolls...

PAGE One screen at a time

HALF Half a screen at a time

DATA One line or column less than a screen at a time

CSR To the cursor position (if the cursor is not positioned on a record when the scroll is performed, then the scroll amount defaults to PAGE)

MAX To the end of the data in the direction indicated by the scrolling function key.

nnnn A number of columns (when scrolling left or right) or a number of records or lines (when scrolling up or down) at a time

Note:

1. When you use the Next function key (F11) or Previous function key (F10) to scroll through records, the only indication that a hidden record has been skipped is that the record number (shown in the upper right corner of the screen) skips accordingly. For details, see "Hiding or showing records in SNGL display format" on page 228.
2. In TABL display format, scrolling left or right moves to the start of the field that is closest to the specified scroll amount. For example, if the **Scroll** field contains CSR, then moving your cursor into a field and pressing the Right function key (F11) scrolls the display right so that the first column of the field is at the left of the screen. If you enter a numeric (*nnnn*) scroll amount, it is treated as an absolute column number (rather than a relative number), and the display scrolls to the start of the field that occupies that column.

You can temporarily override the amount in the **Scroll** field by typing a scroll amount on the Command line, then pressing a scroll function key. For example, if you enter `8` on the Command line, then press the Right function key (F11), File Manager scrolls right 8 columns.

Scrolling to the first or last record or column

To scroll to the first or last record or first or last column in the data set, type `MAX` (or `M`) on the Command line, then press one of the scroll function keys. For example, typing `M` then pressing the Right function key (F11) scrolls right to the last column.

You can also scroll to the first or last record by entering `TOP` or `BOTTOM` on the Command line.

When you perform one of these actions:

- View or edit KSDS or ESDS data sets that do not fit into available memory.
- Use in-place editing for KSDS or ESDS data sets (regardless of the file size).

and you use the BOTTOM, DOWN MAX or NEXT MAX commands (to scroll to the end of the data), File Manager uses the best-performing technique to navigate directly to the last record. Similarly, while viewing a ESDS or KSDS, you may navigate directly to a record by entering a RBA or KEY value. In these situations, as File Manager does not know the relative record number, it displays "Record number unknown" in place of the record number and displays '======' in the prefix area. (This message does not signify an error condition.) To make File Manager aware of the number of records in the data set, issue the RECSTATS primary command.

To reset your display after this has occurred, scroll to the top of the data set.

To avoid this situation, scroll down in increments, for example, `DOWN 9999`, until the end of the data set is reached. Alternatively, if you know the last record number,

use the LOCATE command to go to that record. Once File Manager has recognized the record number, you can scroll in any direction, using any of the various scrolling methods, and this display situation does not reoccur.

Holding fields when scrolling left or right

In TABL display format, when you scroll either right or left, you can edit the template and use the H prefix command to "hold" specific fields, then return to your data view. "Held" fields are displayed at the left of the screen and are always displayed (regardless of how far you scroll).

Related topics

"Holding columns when scrolling left or right" on page 163

Locating a field, column or record number

To scroll to a particular record number, use the LOCATE primary command:

```
LOCATE record_number
```

In all display formats, you can scroll to a particular record number by typing the record number in the **Record** field, then pressing Enter.

In all display formats except SNGL and TABL, you can scroll to a particular column by typing the column number in the **Col** field, then pressing Enter.

In SNGL and TABL display format, you can use the LOCATE primary command to scroll to a particular field, using either the field reference number or the Field Name. For example:

```
LOCATE #5
```

or

```
LOCATE Age
```

When the field is an item in an array, you must specify a subscript in parentheses to identify the occurrence that you want to locate. If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array. For example:

```
L #7(5)
L #9(2,3)
```

In TABL display format, the LOCATE command scrolls your view so that the specified field becomes the leftmost visible field. Once positioned on a field, you can use the Down (F8) or Up (F7) commands (function keys) to scroll through the records and maintain the same relative field view.

In SNGL display format, the LOCATE command scrolls your view so that the specified field becomes the topmost visible field. Once positioned on a field, you can use the Next (F11) or Prev (F10) commands (function keys) to scroll through the records and maintain the same relative field view.

Related topics

"LOCATE primary command" on page 786

Scrollable input and display fields for long names

Some File Manager panels have *scrollable fields* that allow you to input or view longer values than can be displayed at the one time.

There are two types of scrollable fields on File Manager panels:

Input fields
The fields on File Manager panels where you can *enter* information, such as a data set name.

Display fields
The fields on File Manager panels that *display* information as a result of some action you have taken. For example, when you display data on the Browse, Edit, or View panel in a File Manager editor session, the data set name displayed at the top of the panel is a scrollable display field.

The following two sections describe how to work with both types of scrollable field when entering or viewing longer fields.

Working with scrollable input fields

On several panels, File Manager provides scrollable input fields that allow you to enter values longer than length of the displayed entry field.

You can recognise a scrollable field by the presence of one of the following adjacent to the end of the entry field:

+ Plus sign. Indicates the field is scrollable to the right

- Minus sign. Indicates the field is scrollable to the left

+- Plus sign and minus sign. Indicates the field is scrollable to the right or left

If the length of the input field displayed on the panel is insufficient for the value you want to enter, you can either:

- Progressively scroll to the right as you enter the value with the Right function key (F11) or by entering the RIGHT primary command (with the cursor positioned within the entry field), or
- Press the Expand function key (F4) if available, or enter the EXPAND primary command, to display a window that allows you to enter the full length of the value.

  In the expand pop-up window, you can toggle the view of the data between character and hex by issuing the commands HEX OFF and HEX ON.

Deleting data in scrollable input fields: If you want to delete all the data in a field that is displayed as a scrollable input field, take care to ensure that you do, in fact, delete all the data (and not just the displayed data).

If you use the field delete key combination to delete data, File Manager deletes only the data that is currently displayed. If the field contains more data than is displayed, the undisplayed data is left undeleted.

It is recommended that, before deleting data in a field that is displayed as a scrollable field, you first display the entire contents of the field by pressing the Expand function key (F4) or entering the EXPAND primary command. You can then use the field delete key combination to delete all data in the field at once.

Working with scrollable display fields

Some File Manager panels show data displayed in scrollable fields. File Manager uses such fields where the length of the field containing the data to be displayed is longer than the field displayed on the panel.

Scrollable input and display fields for long names

You can recognise a scrollable field (and where more data exists than can be shown at one time on the panel) by the presence of one of the following adjacent to the end of the display field:

+ Plus sign. Indicates the field is scrollable to the right

- Minus sign. Indicates the field is scrollable to the left

+- Plus sign and minus sign. Indicates the field is scrollable to the right or left

Note: The "+", "-", and "+-" signs are only shown adjacent to the end of a scrollable display field when the data to be displayed is longer than the length of the display field on the panel.

If the data to be displayed is longer than the length of the display field on the panel, you can view the rest of the data by either:

- Progressively scrolling to the right by pressing the Right function key (F11) or by entering the RIGHT primary command (with the cursor positioned within the display field), or
- Pressing the Expand function key (F4) or entering the EXPAND primary command to display a window that allows you to view the entire contents of the field.

Related topics
"EXPAND primary command" on page 766

Sorting the data

You can sort your data by using the SORT primary command. With this command, you can sort the data in any of the following ways:

- On up to 5 specified column ranges (CHAR, HEX, or LHEX display formats only).
- On up to 5 fields by specifying the field reference numbers (TABL display format only).
- Into the KEY or SLOT NUMBER sequence for the data set (KSDS, VRDS, and RRDS VSAM data sets).

Note: SORT requires that all data set records be memory-resident. If the entire data set can not fit into your region storage, an error is reported and the SORT execution aborted. If necessary, increase the region size to contain the entire data set.

You can specify whether you want the data sorted in ascending (the default) or descending sequence and, in Edit, you can restrict the sorting to excluded records or not-excluded records only.

To sort by a column range:

1. Display your data in CHAR, HEX or LHEX format.
2. Enter the SORT primary command, followed by up to five pairs of column range values. For example:

   ```
   SORT 56 70 3 22
   ```

   sorts a data set into ascending order, using the range 56-70 as the first key and range 3-22 as the second key.

To sort by field references:

1. Display your data in TABL format (requires that you have specified a template).
2. Enter the SORT primary command, followed by up to five field references. For example:

   ```
   SORT #6 D #3 D
   ```

   sorts a data set into descending order, using field #6 as the first key and field #3 as the second key.

To restore a list to the KEY or SLOT NUMBER order (KSDS, VRDS, and RRDS VSAM data sets):

1. Display your KSDS, VRDS, or RRDS VSAM data set in any display format.
2. Sort the data away from the KEY or SLOT NUMBER order by issuing a SORT command, by moving or adding records or by editing the key field (KSDS only). If you have edited your records, issue the SAVE command.
3. Restore your data to its KEY or SLOT NUMBER order by issuing the SORT command followed by the KEY parameter, for example:

   ```
   SORT KEY.
   ```

   sorts a data set into descending order, using field #6 as the first key and field #3 as the second key.

Examples

- If you are *not* using a template:

  SORT 23 36
  Sorts all records on columns 23 to 36 in ascending order.

  SORT 77 80 D NX
  (Edit only.) Sorts not-excluded records on columns 77 to 80 in descending order (as if excluded records did not exist).

  SORT 77 80 D 23 36 3 21
  Sorts all records on columns 77 to 80 (primary sort key) in descending order, then on columns 23 to 36 (secondary sort key) in ascending order, then on columns 3 to 21 (minor sort key) in ascending order.

  SORT KEY
  Restores the records in a KSDS, VRDS, or RRDS VSAM data set into their original keyed sequence.

  SORT KEY behaves differently for RRDS and KSDS files. For RRDS files the VSAM slot value is used as the sort key while KSDS files use the key field. If new records are added to an RRDS file they are allocated a slot value such that they sort to the end of the file when SORT KEY is performed. New records added to the KSDS are sorted on the new key value and the new records sorted to the appropriate position within the file.

- If you *are* using a template (TABL display format):

  SORT #7 D #6 #3
  Sorts all records on the field with field reference number 7 (primary sort key) in descending order, then on the field with field reference number 6 (secondary sort key) in ascending order, then on the field with field reference number 3 (minor sort key) in ascending order,

SORT #6 X
(Edit only.) Sorts all excluded records on the field with field reference number 6 in ascending order (as if non-excluded records did not exist).

Related topics
"SORT primary command" on page 818

Finding specific data

To find one or more occurrences of a character string in a data set or data set member, you can use the FIND primary command, which you can abbreviate as F or /.

For example, to search for the next occurrence of the string "James" in any mix of uppercase and lowercase, enter the following on the Command line:

```
F JAMES
```

To find the same string as you specified on the previous FIND primary command, specify an asterisk (*) as the search string.

To limit the search for a string to only where it appears as a prefix in the data, specify the PREFIX parameter.

To limit the search for a string to only where it appears as a suffix in the data, specify the SUFFIX parameter.

To limit the search for a string to only where it appears as a "word" in the data, specify the WORD parameter.

Note:

1. To find long strings that do not fit on the Command line, use the FX command.
2. To find and replace strings in multiple data sets or data set members, use the Find/Change Utility.
3. To find and change a string or numeric value, open your data set in an Edit session, then use the CHANGE or CX primary command.
4. By default, in SNGL or TABL display formats, if you enter the CHANGE, FIND, or EXCLUDE command without specifying any field references, File Manager searches all the visible fields based on the template attributes for each field. That is:
   - For a field with alphanumeric field attributes, the command performs a string comparison between the search string and the field being searched.
   - For a field with numeric field attributes, the command interprets the search string as a number, and performs a numeric comparison with the numeric value of the field being searched.
5. If you want to prevent numeric comparisons from occurring, you can explicitly mark the argument as character by using the C'*string*' format. For example, this command finds only instances of alphanumeric fields containing the character "0":

   ```
   FIND c'0' ALL #3,#5
   ```

Related topics
"Handling long strings in CHANGE" on page 107
"Finding and changing data in multiple PDS members" on page 270
"Finding and replacing strings" on page 104

"Setting column ranges" on page 110
"FIND/FX primary command" on page 769
"CHANGE/CX primary command" on page 748

Searching numeric fields

The display format, and whether or not you specify field column numbers (either as a list or a range) affects how the CHANGE, FIND, and EXCLUDE commands search numeric columns:

- When searching data in SNGL or TABL display formats, for a field with numeric field attributes, File Manager by default interprets the search string as a number and performs a numeric comparison with the formatted numeric value of the field being searched.

  The numeric value of the search string must equal the formatted numeric value of the field. The lengths (number of digits) of the search string and of the numeric field are not significant. Similarly, the data type of the numeric field is not significant. For example, a search string of 123 matches a packed-decimal field containing 00123 or a floating-point field containing 1.230E+02. A search string of 123 does not match a packed-decimal field containing 12300 (because the numeric values are different). An unsigned value in the string is considered to be positive.
- In CHAR, HEX, and LHEX display formats, or if you specify a column range, File Manager performs a string comparison between the search string and the character representation of the numeric field. If the search string happens to match the character representations of some of the bytes in a binary or packed decimal numeric field, then the numeric field is considered to be a match for the search string (and, for a FIND command, the entire field is highlighted). Similarly, a search string of 123 matches any "zoned" (or "display") numeric field whose value contains that sequence of digits (for example, 12300 or 41235).

For example, the following command finds instances of an alphanumeric field (#3) containing the character "0" (such as "1060 HAY ST"), and instances of a numeric field (#5) whose formatted numeric value is zero:

```
FIND 0 ALL #3,#5
```

Handling long strings in FIND

You can use the FIND command to specify a string that is up to 100 characters long. However, when you are dealing with long strings, you might find that you cannot fit the whole FIND command (including other parameters) on the Command line. To overcome this problem, do one of the following to display an Extended Command Entry pop-up panel (see Figure 38 on page 86):

- Enter `FX` (for Find Extended)
- Enter the FIND primary command (or one of its abbreviations, such as F) *with no parameters*.

The Extended Command Entry pop-up panel contains five lines (each 50 characters long) that you use to enter all the FIND command parameters (but not the actual keyword FIND or any of its abbreviations). File Manager treats the five lines on the pop-up panel as contiguous, allowing you to specify a long search string and any other parameters, using up to 250 characters in total.

Note: You can continue a string (or any other parameter) from the last position on one line of the pop-up panel to the first position of the next line. However, this does not mean that you need to completely fill a line before using the next line. For example, you can specify the search string on the first of the five lines and the other parameters on the next line. If you are not specifying a quoted string and

you do not use all of a line, File Manager treats the unused part of the line as a single space. When you specify a string with quotes, any trailing blanks and any leading blanks in the next line are retained.

Figure 38 shows the Extended Command Entry pop-up panel for the FIND command. This example searches for all occurrences of the string "dichlorodiphenyltrichloroethane" in field #3 for non-excluded records only.

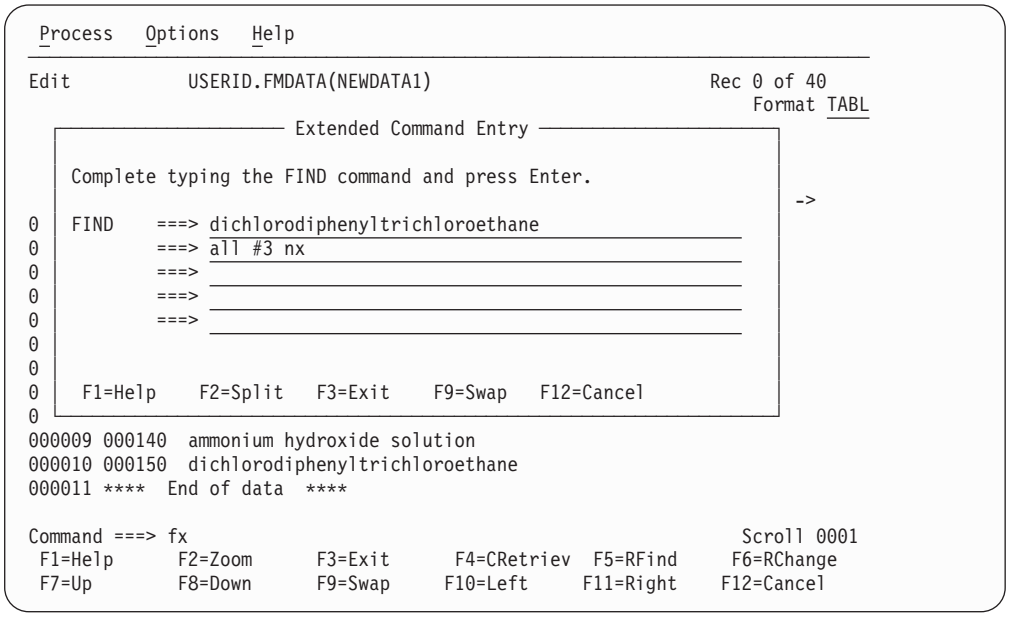


Figure 38. Extended Command Entry pop-up panel for the FIND command

Related topics
- "Handling long strings in CHANGE" on page 107
- "Handling long strings in EXCLUDE" on page 88
- "FIND/FX primary command" on page 769

Finding and correcting errors in your data

To find errors in your data (without correcting them), you can enter the FE (Find Error) primary command. This finds the following types of errors:

- Numeric fields whose contents cannot be formatted as a valid numeric value
- Fields that run over the record length

When you enter the FE command, File Manager positions the cursor at the beginning of the field in error and, if necessary, automatically scrolls the data to bring the field into view.

To move the cursor to the next field in error, either enter the FE command again, or press the RFind function key (F5).

To find *and change* numeric fields flagged as being in error (all asterisks in the field display) to a specific numeric value, use the CE primary command.

Related topics
- "FE (Find Error) primary command" on page 767
- "CE (Change Error) primary command" on page 746
- "Limiting the effect of editing changes" on page 108
- "Changing invalid numeric fields in your data" on page 108

"View panel" on page 707
"Editor panel" on page 534

Excluding records from display

You can hide records from your view, by *excluding* records that have some common factor.

To exclude multiple records based on a common factor:

- Enter EXCLUDE with the relevant parameters on the Command line.

  For example, the following command excludes all records with an "h" or "H" in column 3:

  ```
  EXCLUDE ALL H 3
  ```

  You can control whether the excluded records are completely hidden or are represented by shadow lines.

  – To hide the shadow line for excluded records, enter `SHADOW X OFF` on the Command line The shadow lines disappear, and `SHAD` appears at the top left of the screen, indicating that there are records that are hidden because shadow lines have been turned off.

    You can also hide the shadow line for excluded records by deselecting the **See shadow lines** option, **Excluded**, on the relevant Editor Options panel.

  – To turn on shadow lines for excluded records, enter `SHADOW X ON` on the Command line.

    You can also turn on shadow lines for excluded records by selecting the **See shadow lines** option, **Excluded**, on the relevant Editor Options panel.

EXCLUDE commands are cumulative; each successive EXCLUDE command increases the number of records already excluded.

To "unexclude" (redisplay) excluded records:

- Enter `RESET EXCLUDED` on the Command line.

After you have excluded records, you can limit the effects of the FIND, FE, SHADOW and SORT commands to only those records that are excluded or only those that are not-excluded. You do this by including one of the following option parameters with the primary command:

X or EX
: Affects only excluded records

NX
: Affects only not-excluded records

For example, the following command finds all occurrences of the string "no", "No", "NO", or "nO" in all not-excluded records:

```
FIND NO ALL NX
```

Note:

1. If neither parameter is used with the primary command, both excluded and not-excluded records are affected.
2. If an excluded record is affected by a primary command, either by using the EX parameter or by not specifying an EXCLUDE parameter, the record becomes not-excluded.

Related topics

"EXCLUDE/XX primary command" on page 762

"RESET primary command" on page 806

Handling long strings in EXCLUDE

You can use the EXCLUDE command to specify a string that is up to 100 characters long. However, when you are dealing with long strings, you might find that you cannot fit the whole EXCLUDE command (including other parameters) on the Command line. To overcome this problem, do one of the following to display an Extended Command Entry pop-up panel (see Figure 39):

- Enter XX (for Exclude Extended)
- Enter the EXCLUDE primary command (or one of its abbreviations, such as X) *with no parameters*.

The Extended Command Entry pop-up panel contains five lines (each 50 characters long) that you use to enter all the EXCLUDE command parameters (but not the actual keyword EXCLUDE or any of its abbreviations). File Manager treats the five lines on the pop-up panel as contiguous, allowing you to specify a long search string and any other parameters, using up to 250 characters in total.

Note: You can continue a string (or any other parameter) from the last position on one line of the pop-up panel to the first position of the next line. However, this does not mean that you need to completely fill a line before using the next line. For example, you can specify the search string on the first of the five lines and the other parameters on the next line. If you are not specifying a quoted string and you do not use all of a line, File Manager treats the unused part of the line as a single space. When you specify a string with quotes, any trailing blanks and any leading blanks in the next line are retained.

Figure 39 shows the Extended Command Entry pop-up panel for the EXCLUDE command. This example searches for all occurrences of the string "dichlorodiphenyltrichloroethane" in field #3.

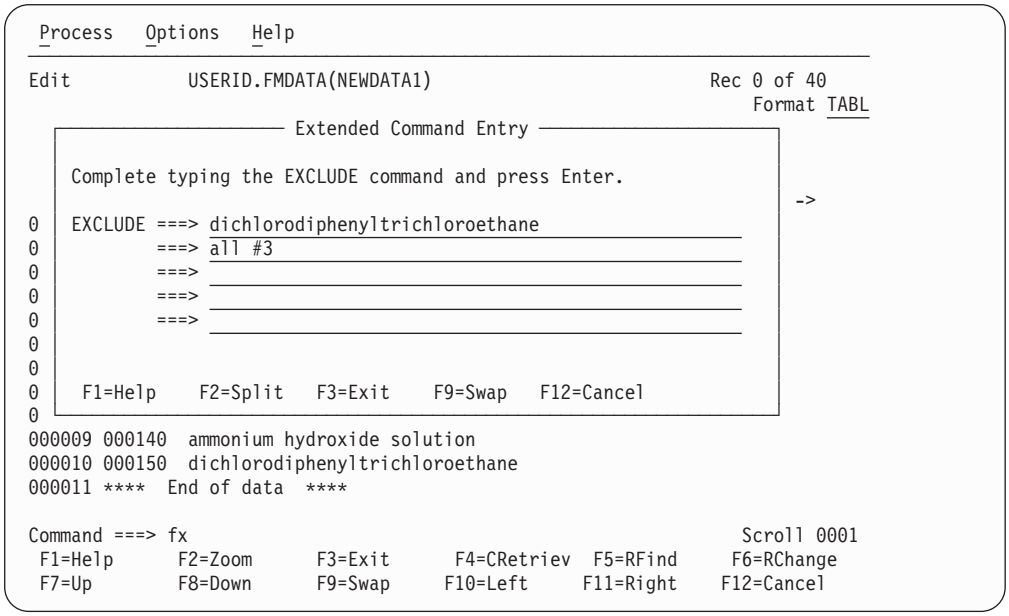


Figure 39. Extended Command Entry pop-up panel for the EXCLUDE command

Related topics

"Handling long strings in CHANGE" on page 107
"Handling long strings in FIND" on page 85

"EXCLUDE/XX primary command" on page 762

Working with File Manager and shared files

This section describes how to work with VSAM files in a shared environment. This may be as a result of using:

- The File Manager for CICS feature, or
- The base File Manager product and accessing a VSAM SHAREOPTIONS 3 or SHAREOPTIONS 4 KSDS file, or another type of VSAM SHAREOPTIONS 3 or SHAREOPTIONS 4 file that could not be allocated exclusively to the user or region accessing it.

  Note: The use of VSAM SHAREOPTIONS 3 files for multiple users where the file may be updated is not recommended as VSAM offers no methods for data integrity for SHAREOPTION 3 files. For this reason, results with such files may be considered unpredictable.

Overview of shared file features

File Manager allows you to work with files that are used in a shared environment.

Features include:

- Ensuring that an update you are about to commit to does not overwrite another user's update (Edit only). This is referred to as *record integrity checking*.
- Ability to refresh the contents of an individual record at any time.
- Refreshing a record before displaying it in SNGL display format.
- Saving an individual record, or range of records, while remaining in the current Edit session.
- Performing periodic SAVEs of a file after a specified number of changes (Edit only).
- While processing a CHANGE command with the ALL operand, performing periodic SAVEs of a file.
- Retrying the CHANGE process for a record that, when being SAVEd, has been updated by another user.

Each of the above features is described later in this section.

What File Manager considers a shared file

For most of the commands discussed in this section, record integrity checking is only active when File Manager considers itself to be operating on a file in a *shared environment*.

A shared environment exists when File Manager is operating on:

- A CICS VSAM file (when you are using the FM/CICS feature), or
- A KSDS VSAM file that is defined with SHAREOPTIONS 3 or 4, or
- A non-KSDS VSAM file that could not be allocated to the region with a DISP of OLD (meaning that exclusive access to the file is not currently available) *and* the VSAM file is defined with SHAREOPTIONS 3 or 4.

Record integrity checking

When File Manager is operating in a shared environment, it enables record integrity checking. This means that, before rewriting the contents of a record to a

file, File Manager first checks if the record has been updated (by another user) since the record was retrieved. File Manager performs this check by comparing the contents of the record at the time that it retrieved it with the contents of the current record as it exists on the file.

With a heavily shared file it is possible that, between the time the record was first fetched from the file and the time that you issued a SAVE command (or a save was automatically issued due to the value of the **Autosave frequency** option, see "Saving a file after a given number of updates (Autosave frequency option)" on page 96), another user updated the same record that you are trying to update.

Saving of records for a file occurs when:

- You issue a SAVE primary command (applies to all records in the file).
- You issue an SV or SVV prefix command (applies to specific records in the file).
- You issue an SV primary command while in SNGL display format (applies to the specific record being viewed).
- The value of the **AUTOSAVE frequency** option triggers a file-wide SAVE.
- The value of the **CHANGE ALL autosave frequency** option triggers a file-wide SAVE during the CHANGE command process.

When File Manager detects that a record has been changed by another user, it is called an *integrity check*.

File Manager recognizes three variations of integrity check:

- You are attempting to update a record but, in the meantime, the record content has been deleted by another user.
- You are attempting to delete a record but, in the meantime, another user has updated it.
- You are attempting to update a record but, in the meantime, the record content has been updated by another user.

In each of the saving scenarios mentioned previously, when an integrity check occurs, File Manager displays one of the integrity check panels shown on the following pages (according to the three variations listed above).

Note: In the case of the saving scenario where the value of the **CHANGE ALL autosave frequency** option (see "CHANGE ALL, automatic saving and retry" on page 97) has triggered a file-wide SAVE during the CHANGE command process, File Manager only displays an integrity check panel if the **Auto retry for CHANGE ALL** option (see "CHANGE ALL, automatic saving and retry" on page 97) is set to OFF.

```
File Manager           Record Updated by other user
Command ===>                                                    Scroll CSR
                                                                Format CHAR
Between the time the record was retrieved and the time a save was issued,
another user has updated the record that you are attempting to update. The
records are shown below and changes highlighted.

Enter  SAVE   to save this record anyway.
Enter  CONTINUE  Save anyway and don't ask again for other records.
Enter  CANCEL to skip this update.
Enter  ABORT  to skip this update and halt performing other updates.

Col 1
Offset      Pending record contents:            Record contents as on file:
       <----+----1----+----2----+----3-->   <----+----1----+----2----+----3-->
000000 00000001A This is my change AAAAAA | 00000001AAAAAAAA meanwhile, user 2
000034 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  |  updated this record AAAAAAAAAAAAA
000068 AAAAAAAAAAAA                      | AAAAAAAAAAAA
```

Figure 40. Integrity check variation 1: (Update attempted but record updated by another user)

```
File Manager           Record Deleted by other user
Command ===>                                                    Scroll CSR
                                                                Format CHAR
Between the time the record was retrieved and the time a save was issued,
another user has deleted the record that you are attempting to update. The
record you are updating is shown below.

Enter  SAVE   to save this record anyway.
Enter  CONTINUE  Save anyway and don't ask again for other records.
Enter  CANCEL to skip this update.
Enter  ABORT  to skip this update and halt performing other updates.

Col 1
Offset      Pending record contents:                  Record deleted
       <----+----1----+----2----+----3-->
000000 00000001AAMaking a change to a rec |
000034 ord that someone else deleted AAAA |
000068 AAAAAAAAAAAA                       |
```

Figure 41. Integrity check variation 2: (Update attempted but record deleted by another user)

```
File Manager           Record Updated by other user
Command ===>                                                    Scroll CSR
                                                                Format CHAR
Between the time the record was retrieved and the time a save was issued,
another user has updated the record that you are attempting to delete. The
records are shown below and changes highlighted.

Enter  SAVE   to save this record anyway.
Enter  CONTINUE  Save anyway and don't ask again for other records.
Enter  CANCEL to skip this update.
Enter  ABORT  to skip this update and halt performing other updates.

Col 1
Offset           Record deleted                 Record contents as on file:
       <----+----1----+----2----+----3-->   <----+----1----+----2----+----3-->
000000 00000001AAAAAAAAAAAAAAAAAAAAAAAAAA | 00000001AAA Another user happily u
000034 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  | pdates the record AAAAAAAAAAAAAAA
000068 AAAAAAAAAAAA                      | AAAAAAAAAAAA
```

Figure 42. Integrity check variation 3: (Delete attempted but record updated by another user)

In each integrity check panel, you have four options that are selectable by tabbing to the keyword and pressing Enter. The options are:

SAVE Ignores the other update and applies your update. Use this option when you are sure that the record should be set the way you have indicated (or deleted).

CONTINUE
Ignores the other update and applies your update. Use this when you are sure that the record should be set the way you have indicated (or deleted). By using CONTINUE, you are instructing File Manager to also automatically SAVE any further records which encounter an integrity check.

Cancel this individual update (PF12 or CANCEL command)
Ignores your update to an individual record element. If you are changing a record, this leaves your changes in your session and marks the record element as having an integrity check.

Abort this and any following updates (ABORT command)
Operates like the Cancel option, but halts the process of the current command so that you can correct or inspect an individual condition. For instance, if integrity checks occur during the End process of an Edit session, and an integrity check panel is displayed, entering ABORT stops the saving process and re-enters the Edit to allow you to inspect records before ending your session again.

Related topics
"SAVE primary command" on page 812
"SV primary command" on page 822

Marking of records that had an integrity check

When an integrity check occurs, such as when you are updating a record but in the meantime another user has updated the same record (the third integrity check variation described in the previous section) then, assuming you CANCELed or ABORTed your change, File Manager marks the record element that caused this situation as having an integrity check.

In multi-line display formats (CHAR, HEX, LHEX, TABL), File Manager places an indicator (=ICHK>) in the prefix area as shown in Figure 43 on page 93.

```
Process   Options   Help
-------------------------------------------------------------------------------
 Edit             FI:FMNKSB1 DS:FMN.RFM0201.KSDS    Rec 0
 Command ===> _____________________________________________________ Scroll CSR
      Refresh on save N                                 Type KSDS    Format TABL
        FIRSTPART LASTTWO TEXTFIELDMAIN                              TEXTMINOR1
               #4      #5 #6                                         #7       +
             ZD 1:6  ZD 7:2 AN 9:42                                  AN 51:15
           <---+->     <-> <---+----1----+----2----+----3----+----4-> <---+----1
 ****** ****  Top of data  ****
 =ICHK>          0       1 A   updated record                        AAAAAAAAAA
 000002          0       2 AAAaaaaaaaaaxAAAAAAAAbAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000003          0       3 AAAAAAAAAAAAAAAAAAAAcAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000004          0       4 AaaaaaaaaaaaaaaaaaaaaaaaaaAAAAAAAAAAAAAA AAAAAAAAAA
 000005          0       5 AAAAAAAAAAAAAAAAAAAAeAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000006          0       6 AAAAAAAAAAAAAAAAAAAAfAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000007          0       7 AAAAAAAAAAAAAAAAAAAAgAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000008          0       8 AAAAAAAAAAAAAAAAAAAAhAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 000009          0       9 AAAAAxAAAAAAAAAAAAAAiAAAAAAAAAAAAAAAAAAA AAAAAAAAAA
 - - -  - - - - - - - - - - - - - - - - - - - - 1 Line(s) not selected
 - - -  - - - - - - REC-TYPE2 - - - - - - - - - - 89 Line(s) suppressed
 000100          1       0 AAA*** updateok & reversed ***AAAAAAAAAAA AAAAAAAAAA
 F1=Help     F2=Zoom     F3=Exit     F4=CRetriev F5=RFind    F6=RChange
 F7=Up       F8=Down     F10=Left     F11=Right    F12=Cancel
```

Figure 43. Example: display of integrity check in TABL display format

In the SNGL display format, File Manager displays a message line as shown in Figure 44.

```
 Process   Options   Help
-------------------------------------------------------------------------------
 Edit             FI:FMNKSB1 DS:FMN.RFM0201.KSDS    Rec 1
 Command ===> _____________________________________________________ Scroll CSR
                                                   Type KSDS    Format SNGL
 The record had been updated by another user. Enter R Top Line is 1    of 5
 Current 01: REC-TYPE1
 Field           Data
 FIRSTPART             0
 LASTTWO             1
 TEXTFIELDMAIN   A   updated record
 TEXTMINOR1      AAAAAAAAAAAAAAA
 TEXTMINOR2      AAAAAxxAAAAAAAA
 ***  End of record  ***









 F1=Help     F2=Zoom     F3=Exit     F4=CRetriev F5=RFind    F6=RChange
 F7=Up       F8=Down     F10=Previous F11=Next     F12=Cancel
```

Figure 44. Example: display of integrity check in SNGL display format

"==MSG>" and "=PROF>" indicators

In order to more effectively describe what has occurred in certain scenarios, File Manager displays an informational message line. If a prefix is being displayed, an indicator ("==MSG>" or, if you have used the PROFILE primary command, "=PROF>") is shown in the prefix area. The line is not displayed in SNGL or ZOOM display format.

File Manager displays an informational message line when:

- When you enter an edit session (expecting to be able to insert and delete) but the dataset is shared, File Manager displays an informational message advising you that the file is shared and that the edit is in inplace mode.
- When ENDing an edit session and an integrity check occurs, you decide not to save, File Manager displays an informational message advising you that the edit session was re-entered.

Related topics
"PROFILE primary command" on page 800

Refreshing the contents of a record

In a highly active environment, records may be updated frequently. On occasions, you may need to reread the contents of a record from the file that you are working on. For example, you may have encountered an integrity check (as previously described), or be waiting for a particular change to occur to a record before continuing on with some other action. In such situations, you can refresh specific records by using either the RF line command or the RF primary command.

For example, assume you are editing a file in SNGL display format as shown in Figure 45.

```
 Process   Options   Help
------------------------------------------------------------------------------
Edit            FI:FMNKSB1 DS:FMN.RFM0201.KSDS    Rec 1
Command ===> _____________________________________________ Scroll CSR
                                                Type KSDS     Format SNGL
                                                Top Line is 1    of 5
Current 01: REC-TYPE1
Field          Data
FIRSTPART            0
LASTTWO          1
TEXTFIELDMAIN   AAAAAAxxzzzzzzzzzzzzzxxxxxxxxxxxxxxxxxxAAAAA
TEXTMINOR1      AAAAAAAAAAAAAAA
TEXTMINOR2      AAAAAAAAAAAAAAA
***  End of record  ***









F1=Help     F2=Zoom     F3=Exit     F4=CRetriev F5=RFind    F6=RChange
F7=Up       F8=Down     F10=Previous F11=Next     F12=Cancel
```

Figure 45.

When you enter the RF line command or the RF primary command and press Enter, File Manager displays the refreshed record (assuming, for this example, that another user has updated the record) as shown in Figure 46 on page 95.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 Edit             FI:FMNKSB1 DS:FMN.RFM0201.KSDS    Re                  RF done
 Command ===> ____________________________________________________ Scroll CSR
                                                    Type KSDS     Format SNGL
                                                    Top Line is 1    of 5
 Current 01: REC-TYPE1
 Field           Data
 FIRSTPART             0
 LASTTWO             1
 TEXTFIELDMAIN   A   updated record
 TEXTMINOR1      AAAAAAAAAAAAAAA
 TEXTMINOR2      AAAAAAAAAAAAAAA
 ***  End of record  ***









 F1=Help     F2=Zoom     F3=Exit     F4=CRetriev F5=RFind    F6=RChange
 F7=Up       F8=Down     F10=Previous F11=Next     F12=Cancel
```

Figure 46.

When you are using multi-line display formats (CHAR, HEX, LHEX, TABL), you can refresh the contents of a record with either the RF or RFF primary command, or the RF prefix command.

Note: If you do have any pending changes for a record, when you refresh the record those changes are discarded without further warning. This can be useful however if, for example, you want to undo your changes to a specific record, or refresh its contents when an integrity check has occurred when attempting to save your change.

Related topics
"RF primary command" on page 808

Refresh of a record in SNGL display format

In SNGL display format, the record is refreshed whenever you navigate to it. That means that the record is read just before being displayed so that you are seeing the most recent version of the record (that it is possible for File Manager to display).

For example, if you use the NEXT command (PF11) followed by the PREVIOUS command (PF10), then you will see any updates made by another user to the original record you were viewing. (This is only true if you do not have any pending changes for the record.)

Related topics
"NEXT primary command" on page 793
"PREVIOUS primary command" on page 798

Saving individual records

File Manager provides commands to save individual records. In multi-line display formats (CHAR, HEX, LHEX, TABL), use the SV or SVV primary command, or the SV prefix command. In SNGL display format, use the SV primary command to save the contents of the record currently being displayed.

Related topics
"SV primary command" on page 822

Saving a file after a given number of updates (Autosave frequency option)

File Manager can perform periodic file-wide SAVEs for shared files. The period or frequency of the SAVE is controlled by the value of the **Autosave frequency** option in the VSAM Edit Sharing Options panel as shown in Figure 47.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager              VSAM Edit Sharing Options


These options apply only when editing VSAM files that are being shared.


 Autosave frequency              ________

Options for the CHANGE ALL command
 SAVE before starting            ___  Enter ON or OFF.
 Change all autosave frequency   ________
 Auto retry for change all       ___  Enter ON or OFF.









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 47. VSAM Edit Sharing Options panel

You can access the VSAM Edit Sharing Options panel by the Options pull-down menu, or by means of the menu options.

You can also disable or enable the **Autosave frequency** option, or specify a frequency value, by using the AUTOSAVE primary command.

What counts as an update?: For the purposes of the **Autosave frequency** option, the following are counted as *one* update:

Overtyping the contents of a record
: Changing the contents of a record by overtyping, regardless of the number of characters changed, is counted as a single change. If you press Enter and then overtype your changed record again, this is counted as a second change.

Using a prefix command that alters record data
: A prefix command that alters data, such as ">" (the command for shift right), is counted as a single change. Use of a prefix command is counted as only one change, even if the command is used to alter multiple records.

The CHANGE or RCHANGE primary command
: When you issue the CHANGE or RCHANGE primary command without the ALL operand, the command counts as one change for each record it actually changes.

Related topics

"CHANGE/CX primary command" on page 748
"RCHANGE primary command" on page 801

CHANGE ALL, automatic saving and retry

On the VSAM Edit Sharing Options panel, there are three options that relate specifically to the CHANGE command when used with the ALL operand:

- **SAVE before starting**
- **Change all autosave frequency**
- **Auto retry for CHANGE ALL**

In a similar way to the **Autosave frequency** option, these options allow the change process to count the number of changes made and perform a periodic SAVE of the file being changed. In addition, the **Auto retry for CHANGE ALL** option is useful for retrying changes to records which encountered an integrity check at the time of the automatic save during the CHANGE command process.

When you select the **CHANGE ALL autosave frequency** option, it is highly recommended that you specify a frequency and also select the **Auto retry for CHANGE ALL** option in order to gain the most benefit from these two features.

For the **CHANGE ALL autosave frequency** option, changes are counted for each record that is changed.

You can also change the setting of the **CHANGE ALL autosave frequency** option and the **Auto retry for CHANGE ALL** option with the CHGASAVE and AUTORTRY primary commands respectively.

When you use these options, be aware that:

- Other changes (such as overtyped records), made before the start of the CHANGE ALL command, are saved either before the CHANGE command commences processing if the **SAVE before starting** option is ON, or at the time of the first automatic save at the CHANGE ALL autosave frequency if the **SAVE before starting** option is OFF.
- Any changes outstanding at the completion of the CHANGE ALL command are saved, regardless of the current change count.
- All records are refreshed at the completion of the CHANGE ALL command.

Related topics

"CHANGE/CX primary command" on page 748
"CHGASAVE primary command" on page 754
"AUTORTRY primary command" on page 741

Editing techniques

When you attempt to edit a data set, File Manager examines your editing options, data set type and size, and uses one of three main variations of editing technique. Although these techniques should be virtually transparent to you, they are explained here.

The type of edit taking place is visible during the editor session by a denotation in the top left of the editor panel title line:

Edit Denotes a full edit (that is, including the ability to insert and delete records, and the ability to copy and move records).

See:

- "In-memory edit" on page 99
- "KSDS edit" on page 100

Aux Edit
Denotes a full edit (that is, including the ability to insert and delete records, and the ability to copy and move records) where File Manager is making use of a secondary file to enable the edit.

See:
• "Auxiliary edit" on page 99

Inplace Edit
Denotes that records may only be updated in place, and that the ability to insert or delete records is not available.

See:
• "In-place-in-memory edit"
• "In-place edit"

If you do not need to insert or delete records, you can improve the performance of the Edit session (especially when working with large files) by selecting the **Inplace edit** option. This avoids the associated delays of copying to and from an auxiliary file.

Note: Unless the text in other sections of this chapter explicitly refers to one of these editing techniques, the information in this chapter applies to all editing techniques. Differences between In-memory, Auxiliary, and In-place editing are highlighted where applicable.

The following sections describe the different editing techniques.

In-place-in-memory edit

This type of In-place edit keeps as many of the records of the data set in virtual storage as will comfortably fit (or as many as specified by a record limit). The data set is updated from these records kept in storage when you enter a SAVE, END, or FILE command.

When File Manager uses this editing technique, it displays "Inplace Edit" in the top left of the editor panel title line.

File Manager uses this type of edit when:

- You have selected **Inplace edit** and the data set being edited is small enough to be loaded entirely into memory, or
- You have specified a record limit of "MEMORY", or
- You have selected **Record Sampling**, or
- You have applied a segmented data template containing selection criteria

In-place edit

This type of In-place edit only keeps records in storage that are currently displayed or that have been modified. The data records are replaced in the data set from the modified records kept in storage when you enter a SAVE, END, or FILE command.

When File Manager uses this editing technique, it displays "Inplace Edit" in the top left of the editor panel title line.

File Manager uses this type of edit when:

- You have selected **Inplace edit** and the data set being edited is too large to be loaded entirely into memory, or

- You have applied a segmented data template

The distinctive thing about both In-place edits, as the name is intended to convey, is that records may only be updated in place. Inserts, deletes, and record length changes are not possible. Note that for VSAM RRDS in In-place mode, the delete and insert commands have a slightly different meaning than normal, in that slots may be marked as empty by the delete command and made active again by the insert command.

Related topics
"Editing large files" on page 100

In-memory edit

With this type of edit, File Manager first tests the available TSO region size and, if the region is large enough, reads the entire data set into memory before displaying the first screen of data. This may cause a noticeable delay whilst loading, but subsequent operations are quicker. As with any usage of large amounts of TSO region, a large File Manager editor region can also have a noticeable effect on other address spaces or cause excessive paging.

When File Manager uses this editing technique, it displays "Edit" in the top left of the editor panel title line.

The data set is completely rewritten from the records kept in storage when you enter a SAVE, FILE, or END command.

File Manager uses In-memory edit when:

- You have *not* selected **Inplace edit**, and
- The data set is a QSAM data set (this may be a member of a PDS or PDSE) or a VSAM REUSE data set, and
- You have not specified a starting point or record limit, and
- The entire data set is small enough to fit comfortably within the available TSO region

Auxiliary edit

Auxiliary edit copies the original data set to a secondary data set. Only records currently being displayed or that have been modified (by change, delete, insert) are kept in storage. File Manager recreates the original data set from the secondary data set and the In-memory changes when you enter a FILE or END command. The SAVE command is not supported when editing a data set using this method. You must end the editor session and re-edit the data set.

When File Manager uses this editing technique, it displays "Aux Edit" in the top left of the Edit panel title line.

File Manager uses Auxiliary edit when you have *not* selected **Inplace edit**, or you selected to edit an HFS file, or you selected to edit an SMS-compressed data set, and:

- There is insufficient TSO region for In-memory edit, or
- You have specified a starting position or record limit

The auxiliary data set created by File Manager uses your TSO ID as the prefix High Level Qualifier. Your site may have its own standards and naming

conventions for these data sets which result in the assignment of different storage categories (for example, classes, units, or privileges).

Your FM Administrator can change this default in the customization and installation process. To override this default:

1. From the Primary Options menu, select option 0.8 to display the Set Temporary Data Set Allocation Options panel.
2. Enter a name pattern in the **High Level Qualifier** field. The pattern can be any multi-level qualifier, up to 24 bytes, and can include the following symbols:

   &USER
   Represents User ID

   &PREFIX
   Represents TSO prefix

   Multi-level qualifiers that include symbols follow the same rules as used in JCL procedures for data set names with symbolic parameters. For example, assuming &USER=XXXX and &PREFIX=YYYY, the following HLQs could be set:

   `&USER.1.TEMP.&PREFIX` becomes XXXXl.TEMP.YYYY

   `FMN.&USER..ABCO` becomes FMN.XXXX.ABCO

   The temporary HLQ serves as a user-created prefix for data sets created by File Manager. These data sets have fixed names which can vary in length. The maximum system limit for a fully-resolved data set name (that is, the HLQ and the fixed name) is 44 bytes. If the combination of the HLQ and the fixed name resolve to more than 44 bytes, the least important part of the HLQ (the rightmost level) is ignored.

   Each level can only be 8 bytes long. If a construction such as &USER.ABCD resolves to more than 8 bytes, the rightmost part of the string is ignored.
3. If your site uses the Storage Management System (SMS), enter values in the **Data class**, **Storage class**, and **Management class** fields.
4. Press F3 to save the settings and exit from the panel.

Related topics
"Editing large files"
"Editor Options panel" on page 546

KSDS edit

A VSAM KSDS file has, by its nature, the ability to insert and delete records, without requiring them to be in sequence (as they are sequenced by VSAM in their key order). Hence applications intending to add, change, or remove records (such as File Manager) do not need to keep the entire file contents available (in order to rewrite the entire file) when saving changes. So, for KSDS files, File Manager only keeps new, changed, or deleted records in storage (as well as those currently being displayed). So it is possible for File Manager to edit a KSDS file of any size, without needing to employ the auxiliary edit technique described earlier.

When File Manager uses this editing technique, it displays "Edit" in the top left of the editor panel title line.

Editing large files

Using Auxiliary edit or In-place edit, you can edit large files, regardless of their size, without the need to first break them up into smaller subsets.

If you do *not* select **Inplace edit** on the editor entry panel and perform either of these actions:

- You attempt to edit a QSAM data set or a VSAM ESDS or RRDS data set defined with the REUSE attribute but it is too large to fit into available memory,
- You attempt to edit a QSAM data set or a VSAM ESDS or RRDS data set defined with the REUSE attribute, and you specify a start position or record limit

File Manager uses an auxiliary file in order to fulfill your request and issues progress messages periodically during the pre and post edit copy process.

Editing load modules

You can edit load modules located in a PDS(E) using File Manager. The edit method defaults to in-place so you cannot delete or insert records.

Note: Special authority may be required for this function, depending on the installation options. For more information, see the *File Manager Customization Guide*.

Changing data

When editing a data set, you can change the data:

- In individual records by overtyping the data within the record or by entering a prefix command in the prefix area next to each row. This method is useful when you need to make ad hoc changes to specific records.
- In multiple records by entering primary commands on the Command line. This method is useful when you need to apply global changes or changes to groups of records that have a common denominator.

  The effects of these changes can be limited by using command parameters or by applying limiting methods (see "Limiting the effect of editing changes" on page 108).

  The tasks described in this section are:
  - "Changing data in individual records"
  - "Change data in multiple records" on page 102
  - "Locating labeled records" on page 103
  - "Changing the case of data" on page 103
  - "Finding and replacing strings" on page 104

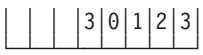
Changing data in individual records

To change the data in an individual record:

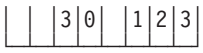
1. Move your cursor to the position where you want to start editing the text.

   This can be done with your mouse or arrow keys or by using the LOCATE or FIND commands.
2. Overtype the existing text with your changes or insert or delete characters.
3. When you change a *numeric* field by overtyping the existing contents, File Manager performs the following steps to determine the new value of the field:
   a. Scans the contents of the field from left to right.
   b. Ignores leading blanks
   c. Uses the numeric digits present in the field up to either the end of the field, or up to the first embedded blank.
   d. Ignores any characters to the right of an embedded blank.

For example, File Manager treats the following 8-character numeric field as having the value, 30,123:



but treats the following 8-character numeric field as having the value, 30:



4. Press Enter (or any function key except Help (F1) or Cancel (F12)).

   Your changes are saved in memory but not written to the file, and the changed record becomes highlighted. You can change more than one record at a time before entering the changes. Your accumulated record changes are only written to file when you issue a SAVE, FILE, END or EXIT command.

Change data in multiple records

When you are *editing* data, you can change data in multiple records, by entering a primary command. Here are the commands, listed in functional groups:

Action Command

Limit the column range affected by the CHANGE, EXCLUDE and FIND commands.
"BOUNDS primary command" on page 743

Find and replace a character string (or a numeric value in a numeric field)
"CHANGE/CX primary command" on page 748

Change data in error
"CE (Change Error) primary command" on page 746

Split or join records
"JOIN primary command" on page 782
"SPLT primary command" on page 821
"SPLTJOIN primary command" on page 821

Save your edited data or cancel without saving
"CANCEL primary command" on page 744
"SAVE primary command" on page 812
"FILE primary command" on page 769

Control the display of the prefix area
"PREFIX primary command" on page 798

Cancel pending prefix commands, remove record labels, display excluded records, and remove highlighting from found strings.
"RESET primary command" on page 806

Exclude certain records from display
"EXCLUDE/XX primary command" on page 762

Delete data, and recover deleted data
"DELETE primary command" on page 759
"Recovering deleted records" on page 117

Copy data within the same record
"COPY primary command" on page 755

Limit the columns to be edited
"Setting bounds" on page 109

Translate data to uppercase
"Changing the case of data"
"Changing the case of data"

Differences between primary commands when editing and viewing

These primary commands, available in both View and Edit, have the following additional features in Edit:

CANCEL, END and QUIT
In View, CANCEL, END and QUIT are synonymous. In Edit, END automatically saves changes before ending the editor session, while CANCEL and QUIT end the editor session without saving changes; if you have made any changes, File Manager displays a message box asking you to confirm that you want to discard those changes.

Locating labeled records

You can use LOCATE to move to a record to which a particular label has been assigned. For example, to move to the record labeled ".HERE", enter the following command:

```
LOCATE .HERE
```

For more information about creating and using labels, see "Labelling record ranges" on page 113. For more information about using the LOCATE command, see "Finding specific data" on page 84.

Changing the case of data

You can translate data to uppercase on input, or you can change the case of existing data to upper or lowercase.

The **CAPS initially ON - translate changed data to uppercase (CAPS)** option on the relevant Editor Options panel determines the initial setting of the "convert to uppercase" feature. Select this option if you normally want data entered in input fields converted into uppercase. Do not select this option if you want to enter lowercase or mixed-case data. This option also affects the processing of CHANGE arguments.

The CAPS and CASE primary commands control whether data in the data area is translated to uppercase on input (CAPS, CAPS ON or CASE UPPER), or left alone (CASE, CASE MIXED or CAPS OFF. As already mentioned, the initial setting of the "convert to uppercase" feature at the start of an editor session is determined by whether the **CAPS initially ON - translate changed data to uppercase (CAPS)** option on the Editor Options panel is selected or not.

The CASE UPPER, CAPS or CAPS ON commands are similar to the ISPF editor CAPS command. If you enter any of these commands, or the **CAPS initially ON - translate changed data to uppercase (CAPS)** option is selected (and no CAPS or CASE commands have been used in the current File Manager session):

- If you overtype any data in a record, then:
  - In CHAR, HEX or LHEX display format, the entire record is translated to uppercase.
  - In SNGL or TABL display format, only the current field is translated to uppercase.

- The CHANGE command treats the new (replacement) string as if it were specified in all uppercase except in these case:
  - The new string is enclosed in quotation marks, prefixed by the letter C (for example, `C'New String'`)
  - The new string is specified as a hexadecimal string (for example, `X'C1C2C3'`)

  If either of the above is true, then the new string is not translated to uppercase.

The CAPS OFF, CASE, and CASE MIXED commands have the same effect. If you enter any of these commands, or the **CAPS initially ON - translate changed data to uppercase (CAPS)** option is unselected (and no CAPS or CASE commands have been used in the current editor session):

- Translation to uppercase is turned off.

Any CAPS command overrides any previously entered CASE command, and any CASE command overrides any previously entered CAPS command.

Note: In the ISPF editor with CAPS ON, a line is considered changed if you type *anything* on the line (if you overtype a space with a space, it is considered to be a change). However, in a File Manager editor session, overtyping a character with the same character is not considered a change.

You can change the case of existing data to uppercase or to lowercase, without having to input new data, by using prefix commands.

Caution: These prefix commands (LC, LC*n*, LCC, UC, UC*n*, UCC) affect all characters in a record, not just characters in those fields with an alphanumeric or character data type. This means that numeric data, such as binary data or packed decimal data, can be corrupted by using these commands.

To change existing data to uppercase:

1. Position your cursor in the prefix area of the record that you wish to change.
2. Type the UC or UC*n* prefix command, where *n* represents the number of records to be changed, or type UCC on the first of a block of records to be changed and UCC in the prefix area of the last record in the block.
3. Press Enter. The records are changed to uppercase.

To change existing data to lowercase:

1. Position your cursor in the prefix area of the record that you wish to change.
2. Type the LC or LC*n* prefix command, where *n* represents the number of records to be changed, or type LCC on the first of a block of records to be changed and LCC in the prefix area of the last record in the block.
3. Press Enter. The records are changed to lowercase.

Finding and replacing strings

To find and replace one or more occurrences of a character string in a data set or data set member, you can use the CHANGE primary command. In SNGL or TABL display format, you can also use the CHANGE command to find and replace a numeric value in numeric fields.

Note:

1. To find and replace long strings that do not fit on the Command line, use the CX command (see "Handling long strings in CHANGE" on page 107).

2. To find and replace strings in multiple data sets or data set members, use the Find/Change Utility (see "Finding and changing data in multiple PDS members" on page 270).
3. To find a string or numeric value without changing it, use the FIND primary command. See "FIND/FX primary command" on page 769.

To change character strings with the CHANGE command:

1. Display your data in the editor panel (see "Starting and ending editor sessions" on page 51 for details).
2. If you are using a template and wish to limit the effects of the CHANGE command, apply suitable record identification or record selection criteria to select your records for processing (see "Selecting records with templates" on page 203).
3. If you want to prevent particular records from being affected by the CHANGE command, use commands such as BOUNDS or EX to limit the range of the CHANGE command (see "Limiting the effect of editing changes" on page 108).
4. On the Command Line, enter the CHANGE command, together with any desired parameters, and the search and replace strings. For example:

   ```
   CHANGE black white
   ```

   See "CHANGE/CX primary command" on page 748 for details about the syntax and parameters of the CHANGE command.

   The CHANGE command highlights all occurrences of the search string or numeric value in the entire data set.

   To turn off the highlighting, enter the RESET FIND command.

   If the CHANGE command changes data, then File Manager places the cursor at the start of the changed data; if necessary, automatically scrolling to bring the data into view.

   To change the next occurrence of the string, press the RChange function key (F6).

   To find the next occurrence of the string and optionally change it, use a combination of the RFind function key (F5) and the RChange function key (F6).

   To find the same string as you specified on the previous CHANGE primary command, specify an asterisk (*) as the search string.

   To use the same replacement string as you specified on the previous CHANGE primary command, specify an asterisk (*) as the replacement string.

   To limit the search for a string to only where it appears as a prefix in the data, specify the PREFIX parameter.

   To limit the search for a string to only where it appears as a suffix in the data, specify the SUFFIX parameter.

   To limit the search for a string to only where it appears as a "word" in the data, specify the WORD parameter.

   To change all occurrences of the string (except those in not-selected or suppressed records), use the ALL parameter with the change command.

The CHANGE command *does not* affect not-selected or suppressed records that are hidden or represented by shadow lines. To include these records in a CHANGE command, you need to show them. For details, see "SHOW primary command" on page 817.

The CHANGE command *does* affect records that have been hidden using the EXCLUDE command, unless you have used the NX parameter. If a change affects an excluded record, it becomes not-excluded. For more information on using the

EXCLUDE command or EX and NX parameters, see “Excluding records” on page 112, “CHANGE/CX primary command” on page 748 or “EXCLUDE/XX primary command” on page 762.

If you have zoomed in on a record, then the CHANGE command affects only that record. See “Zooming in to see all of a record” on page 71.

The CHANGE command may affect record length:

- When you use the CHANGE command in CHAR, HEX or LHEX display format:
  - For variable-length files where the edit type supports record length changes, if *string2* is longer than *string1* and there are not enough EBCDIC blank characters (X'40') at the end of the record to accommodate the difference in length, File Manager expands the record length as required.
  - For variable-length files where the edit type supports record length changes, if *string2* is shorter than *string1*, then File Manager reduces the record length as required.

  Otherwise, for fixed-length files, or where the type of edit prevents record length changes:
  - If *string2* is longer than *string1*, then the change only occurs if there are enough EBCDIC blank characters (X'40') at the end of the record to accommodate the difference in length. File Manager maintains the same record length by dropping blanks from the end of the record.
  - If *string2* is shorter than *string1*, then File Manager maintains the same record length by padding the end of the record with blanks.
- If you use the CHANGE command in SNGL or TABL display format to change a non-numeric field, the change is the same as the above fixed-length considerations except it applies to a field rather than a record. When record length changes are possible, and the field being affected by the change overlaps the end of the underlying record, File Manager may expand the record as required to complete the change request. This only occurs when the underlying record is shorter than the record structure.

  If you use the CHANGE command in SNGL or TABL display format to change a numeric field, then the length of *string1* and *string2* are not significant. For example, if field #2 is a numeric field, then the following CHANGE command is allowed, because *string1* and *string2* are interpreted as numeric values:

  ```
  CHANGE 107 2 #2
  ```

  For more information on changing values in numeric fields, see “Searching numeric fields” on page 85.
- If the argument is a DBCS string, only data between a shift-out and shift-in is searched for a match. For example, a DBCS blank `X'0E40400F'` does not find two consecutive blanks in non-DBCS data.

  The exception to this rule is in TABL or SNGL format where the field is a graphic field. Unformatted graphic fields are not treated as DBCS.

For details on limiting the columns, fields or records searched by the CHANGE command, see “Limiting the effect of editing changes” on page 108.

Examples

- If you are *not* using a template:

CHANGE CAT DOG
When zoomed in, changes the next occurrence of “CAT” to “DOG” in the current record only. Otherwise, changes the next occurrence of “CAT” to “DOG”.

CHANGE YELLOW ORANGE ALL
When zoomed in, changes all occurrences of “YELLOW” to “ORANGE” in the current record only. Otherwise, changes all occurrences of “YELLOW” to “ORANGE” in all records.

- If you *are* using a template, and the data is displayed in SNGL or TABL format:

CHANGE BLACK WHITE (#4 #7)
In TABL format, changes the next occurrence of “BLACK” to “WHITE”, where “BLACK” is completely contained within either of the fields represented by field references 4 and 7. In SNGL format, changes the next occurrence of “BLACK” to “WHITE”, where “BLACK” is completely contained within either of the fields represented by field references 4 and 7, in the current record only.

CHANGE BLACK WHITE ALL #2,#8
In TABL format, changes all occurrences of “BLACK” to “WHITE”, where “BLACK” is completely contained within either of the fields represented by field references 2 and 8, in all records. In SNGL format, changes all occurrences of “BLACK” to “WHITE”, where “BLACK” is completely contained within either of the fields represented by field references 2 and 8, in the current record only.

Handling long strings in CHANGE

You can use the CHANGE command to specify “from” and “to” strings up to 100 characters long. However, when you are dealing with long strings, you might find that you cannot fit the whole CHANGE command (including other parameters) on the Command line. To overcome the problem, do one of the following to display an Extended Command Entry pop-up panel (see Figure 48 on page 108):

- Enter `CX` (for Change Extended)
- Enter the CHANGE primary command (or one of its abbreviations, such as C) *with no parameters*.

The Extended Command Entry pop-up panel contains five lines (each 50 characters long) that you use to enter all the CHANGE command parameters (but not the actual keyword CHANGE or any of its abbreviations). File Manager treats the five lines on the pop-up panel as contiguous, allowing you to specify long “from” and “to” strings and any other parameters, using up to 250 characters in total.

Note: You can continue a string (or any other parameter) from the last position on one line of the pop-up panel to the first position of the next line. However, this does not mean that you need to completely fill a line before using the next line. For example, you can specify the “from” string on the first of the five lines, the “to” string on the next line, and the other parameters on the next line. If you are not specifying a quoted string and you do not use all of a line, File Manager treats the unused part of the line as a single space. When you specify a string with quotes, any trailing blanks and any leading blanks in the next line are retained.

Figure 48 on page 108 shows the Extended Command Entry pop-up panel for the CHANGE command. In this example, all occurrences of the string “dichlorodiphenyltrichloroethane” in field #3 are to be changed to the string “trinitrophenylmethylnitramine” for non-excluded records only.

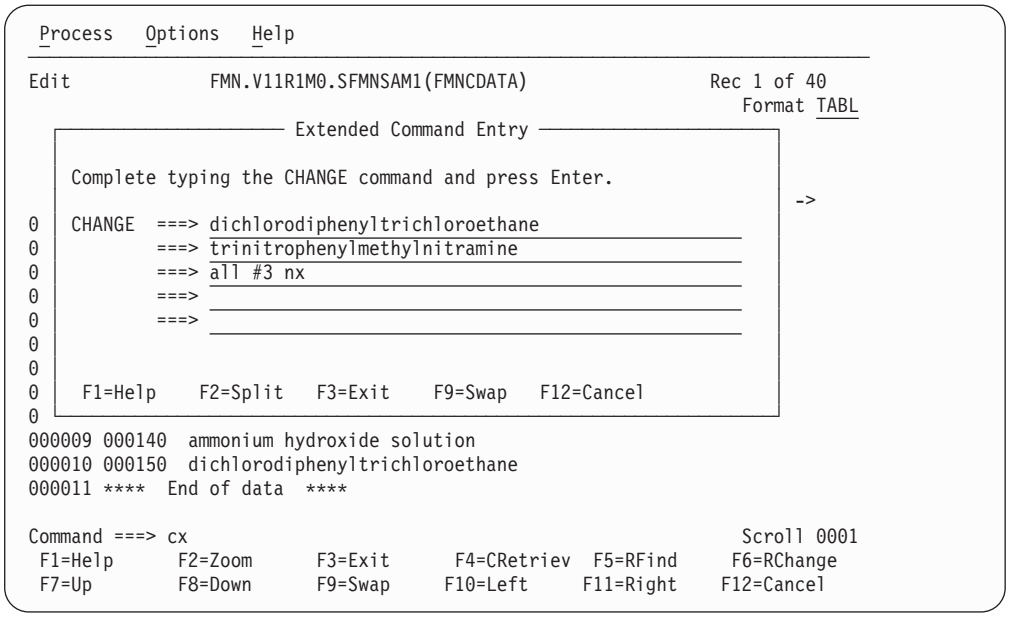


Figure 48. Extended Command Entry pop-up panel for the CHANGE command

Related topics
- "Handling long strings in EXCLUDE" on page 88
- "Handling long strings in FIND" on page 85
- "CHANGE/CX primary command" on page 748

Changing invalid numeric fields in your data

To change an invalid numeric field in your data (all asterisks shown in the field display) to a specified numeric value, you can enter the CE (Change Error) primary command.

The find argument for the CE command is assumed to be a numeric field in error.

When a numeric field in error is found, it changes the field to the string specified in the CE command. The specified string must, of course, be a valid numeric value.

To change the next field in error (to the same value), either enter the CE command again, or press the RChange function key (F6).

To just *find* numeric fields flagged as being in error (without changing them), use the FE primary command.

Related topics
- "CE (Change Error) primary command" on page 746
- "FE (Find Error) primary command" on page 767
- "Limiting the effect of editing changes"
- "Editor panel" on page 534

Limiting the effect of editing changes

When you use prefix commands to edit data, the changes made are applied to the current record only. When you use primary commands to change data, the changes may be applied to the entire data set. While many of these commands have parameters that allow you to limit the effect of the command, there are several

additional methods that you can use to limit the effect of primary commands to only those records that you want to change.

These methods are:

Limiting method	CHANGE	DELETE	EXCLUDE	FIND
Bounds Set left and right column bounds.	✓		✓	✓
Column ranges Override the bounds by specifying column numbers on the command.	✓		✓	✓
Field list[1] Use #*n* field references to specify the fields to be searched.	✓		✓	✓
Excluding records Use only excluded records, not-excluded records, or both.	✓			✓
Record ranges Use record labels to specify a record range.	✓	✓	✓	✓

Note:

1. SNGL and TABL display formats only.

The tasks described in this section are:

- "Setting bounds"
- "Setting column ranges" on page 110
- "Limiting the search to specified fields" on page 111
- "Excluding records" on page 112
- "Labelling record ranges" on page 113
- "Searching numeric fields" on page 85

Setting bounds

In CHAR, HEX and LHEX display formats, you can limit the columns that are searched by the CHANGE, EXCLUDE and FIND commands by setting left and right bounds. There are two methods of setting these bounds.

Method 1:

1. On the Command line, enter

   ```
   BOUNDS col1 col2
   ```

   (see "BOUNDS primary command" on page 743 for information about the parameters of the BOUNDS command).

Note: This method is very quick but does not give you any visual feedback as to where the boundaries lie.

Method 2:

1. In the prefix area of any record, enter `BND`.

A new row is inserted into the display, showing a left angle bracket (<) and, if previously set, a right angle bracket (>).

```
File Manager Edit USERID.FMDATA(DATA1)
       Col 1          Insert length 7996                          Format CHAR
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
000025 02Tyrone Dallas       Developer     92  Smith Street    Belmont
BND    <                             >
000026 02Rod Turnbull        Manager       184 Alexander Dve   Swan View
```

If no bounds have previously been set, the left bracket appears in the first column of your records. If bounds have already been set, the left and right brackets appear in their pre-set columns.

2. Change the bracket positions to suit your needs, by overtyping existing brackets and typing in a left and, optionally, a right bracket.
3. Press Enter or use a function key to scroll through your data. The bounds are updated.

Note: You can display as many bounds lines as you want, which allows you to insert them at convenient locations while you are editing. When the Enter key is pressed, the most recently changed bounds line is processed to reset the boundaries. If you change more than one bounds line before pressing Enter, the lowermost of the changed bounds lines is processed.

Regardless of which method you used to set the bounds, you can reset the bounds to their default by entering the BOUNDS command without any parameters (equivalent to `BOUNDS 1 max_record_length`).

To remove a bounds line from the display, enter `D` in the prefix area of the line. To remove all bounds lines from the display, enter RESET in the Command line.

Note: Removing the bounds lines from the display does not remove the actual bounds.

Setting column ranges

By default, in SNGL or TABL display formats, if you enter the CHANGE, FIND, or EXCLUDE command without specifying any field references, File Manager searches all the visible fields based on the template attributes for each field.

You can override this method of searching by specifying a pair of column numbers, indicating the first and last columns to be searched when you enter the primary command.

Note: When you specify a column range, all fields in the record data are searched as if they were character fields. For numeric fields, this can cause unexpected results.

The search string is found if it is completely contained within the specified columns.

The following command finds the next record, excluded or not-excluded, containing the string “Exact” within columns 1 to 20:

```
FIND C'Exact' 1 20
```

(For an explanation of excluded and not-excluded records, see “Excluding records” on page 112.)

The following command changes all occurrences of the string “no”, “NO” or “nO” to “No” in columns 21 to 30 in all not-excluded records:

```
CHANGE NO C'No' ALL 21 30 NX
```

If the second column that you specify is larger than the record size, then the record size is used. If you specify a single column, then the character string must start in the specified column.

Specifying either a single column or a pair of columns overrides the current bounds. If you do not specify any columns then, when in CHAR, HEX or LHEX display formats, the columns to be searched default to the current bounds (see “Setting bounds” on page 109).

Related topics
“Finding specific data” on page 84

Limiting the search to specified fields

In SNGL or TABL display format, you can optionally limit the effects of primary commands to a specified field, a list of fields, or one or more ranges of fields.

You specify the fields to be searched using #*n* field references. These are displayed above each field in TABL display format, or next to each field in SNGL display format. The search string is found only if it is completely contained within one of the specified fields.

Specifying a list of fields

When you specify a list of fields, you must specify them adjacent to each other. You can optionally enclose the field references (optionally separated by commas) in parentheses.

These are valid lists of fields:

```
#17
(#17)
(#17 #22)
(#17, #22)
#17,#22
#17 #22
#17, #22
```

The search string is found if it is completely contained within one of the specified fields. For example, the following command excludes all records containing the characters “the” (in any mix of uppercase and lowercase) in field #5:

```
EXCLUDE ALL the #5
```

The following command finds the next record containing an uppercase letter A in field #8 or #9:

```
FIND C'A' #8,#9
```

Fields that are items of a multi-dimensional array share the same field reference number, but are differentiated by the dimension, using a subscript. For example, given the following field references (shown in TABL display format):

```
(1,1)    (1,2)    (1,3)    (2,1)    (2,2)    (3,2)
#9       #9       #9       #9       #9       #9
```

You could specify the following lists of fields:

```
FIND bike #9(1,2)
FIND bike (#9(1,3) #9(2,2))
```

To search *all* items in the array, specify:

```
FIND bike #9
```

Specifying ranges of fields

You specify a range of fields by nominating the first and last field reference of the range, separated by a hyphen, with or without intervening spaces. You can optionally enclose the range in parentheses.

These are valid ranges of fields:

```
#17-#19
#17 - #19
(#17-#19)
(#17 - #19)
```

To specify more than one range of fields:

- Enclose each field range in parentheses (optionally separated by commas), or
- Separate each field range with commas.

These are valid ranges of fields:

```
(#8-#11) (#17-#19) (#24-#25)
(#8-#11),(#17-#19),(#24-#25)
#8-#11,#17-#19,#24-#25
#8-#11 #17-#19 #24-#25
#8-#11, #17-#19, #24-#25
```

Note:

1. When using field ranges, the field reference number can not be subscripted. For example, `#3(2) - #5(4)` is not valid syntax.
2. Field ranges can overlap. For example, `#2-#7, #5-#9`. In this case, the result is a consolidated single field range equivalent to `#2-#9`.

Excluding records

You can limit the records affected by the FIND, CHANGE and DELETE commands, first by excluding some records from display, then by limiting those commands to affecting only the not-excluded records, or only the excluded records.

To exclude individual records on an ad hoc basis, choose one of the following methods:

- Enter `X` in the prefix command area of the record to be excluded.
- Enter `X`*n* in the prefix command area of the first record of *n* records to be excluded.
- Enter `XX` in the prefix command area of the first record of a block of records to be excluded, then enter `XX` in the last record to be included in the block.

To exclude multiple records based on a common factor:

- Enter EXCLUDE with the relevant parameters on the Command line (see "EXCLUDE/XX primary command" on page 762).

  For example, the following command excludes all records with an "h" or "H" in column 3:

  ```
  EXCLUDE ALL H 3
  ```

EXCLUDE commands are cumulative; each successive EXCLUDE command increases the number of records already excluded.

To "unexclude" (redisplay) excluded records, choose one of the following methods:

- To unexclude a single excluded record, enter `F` or `L` in the prefix command area of the excluded line.
- To unexclude the first *n* records in a block of excluded records, enter `F`*n* in the prefix command area of the excluded line.
- To unexclude the last *n* records in a block of excluded records, enter `L`*n* in the prefix command area of the excluded line.
- To unexclude all excluded records, enter `RESET EXCLUDED` on the Command line (see "RESET primary command" on page 806).

After you have excluded records, you can limit the effects of the FIND, CHANGE and DELETE commands to only those records that are excluded or only those that are not-excluded. You do this by including one of the following option parameters with the primary command:

X or EX
: Search only excluded records

NX
: Search only not-excluded records

For example, the following command changes all occurrences of the string "no", "NO" or "nO" to "No" in all not-excluded records:

```
CHANGE NO C'No' ALL NX
```

Note:

1. If neither parameter is used with the primary command, both excluded and not-excluded records are affected.
2. If an excluded record is affected by a primary command, either by using the EX parameter or by not specifying an EXCLUDE parameter, the record becomes not-excluded.

Labelling record ranges

In any display format except SNGL, you can limit the number of records affected by the CHANGE, DELETE, EXCLUDE and FIND commands by specifying a record range. A record range consists of a pair of record *labels* indicating the first and last records to be searched. The string is found if it is contained in a record within the specified range.

To limit editing primary commands using your own labels:

1. Enter a label in the prefix area of one data record.

   A record label consists of a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with ".Z" are reserved for the editor (see "Editor-assigned labels" on page 115).
2. Enter a label in the prefix area of a second data record.

   The order in which you specify the labels is not important.
3. Enter your primary command and any parameters, followed by two labels in any order.

When you specify a record range, you must separate the two labels by a blank or comma. A single label is invalid. The labels may be any combination of the editor-assigned labels (.ZCSR, .ZFST, .ZLST) and labels you have assigned yourself.

Regardless of the order in which you specify the two labels, the label representing the smaller relative line number is used as the start of the range and the label representing the larger relative line number is used as the end of the range.

Your cursor is positioned on the first instance that occurs within the range.

Note:

1. When using FIND or CHANGE, all instances of your search string are highlighted, however, only those instances within the range are affected by the command or by the RFIND (F5) and RCHANGE (F6) commands.
2. Once assigned, a label stays with the record, even if the record is moved.
3. A label is removed when the record containing the label is deleted.
4. Overtyping a label with a prefix command does not remove the label. When the command is complete, the record label reappears.
5. You can move a label to another record by typing the same label on a new record.
6. If the range you specify includes not-selected or suppressed records that are hidden from display or represented by shadow lines, then those records *are not* affected by the CHANGE, DELETE, EXCLUDE or FIND command.
7. If the range you specify includes excluded records and you either use the EX parameter or do not specify an EXCLUDE parameter, the records *are* affected by the primary command. If an excluded record is affected by a primary command, the record becomes not-excluded.

In the following example, the label .HERE is assigned to record number 16 and the label .THRE is assigned to record number 129:

```
⋮
00015 This is a data record.
.HERE This is a data record with a label.
00017 This is a data record.
⋮
00128 This is a data record.
.THRE This is a data record with a label.
00130 This is a data record.
⋮
```

This range can now be used to limit a command:

```
CHANGE A B .HERE .THRE
```

The following command excludes all records, excluded or not-excluded, containing the string “where” in the range of records from the label .HERE to the label .THRE:

```
EXCLUDE C'where' ALL ABC .HERE .THRE
```

To remove labels from records, do one of the following:

- For a single label: overtype the label with spaces (or a new label), then press Enter.
- For all labels: enter the RESET LABEL command.

Editor-assigned labels

Some labels are automatically assigned and maintained by the editor. These special labels begin with the letter "Z". (Your own labels, therefore, must not start with a "Z".) Unlike other labels, these editor-assigned labels do not necessarily stay with the same record, but instead represent logical positions on the display. You can use the following editor-assigned labels:

.ZCSR To the record on which the cursor is currently positioned.

To use this label:

1. Type the command and labels on the Command line but do not press Enter.
2. Position your cursor anywhere in the record that you want to use as the current position.
3. Press Enter. All records that lie between your cursor position and the other label are included in the range, regardless of the order in the labels occur in the data set or in which you specify the labels.

.ZFST To the first record (relative record number 1). This can be abbreviated to .ZF.

.ZLST To the last record. This can be abbreviated to .ZL.

The following command changes the first occurrence of the string *xxx* in the range of records from the current cursor position to the end of data:

```
CHANGE xxx yyy .ZCSR .ZLST
```

The following command finds the next record, excluded or not-excluded, containing the string *xxx* in the range of records from the current cursor position to the end of data:

```
FIND xxx .ZCSR .ZLST
```

Editing records within data sets

As well as editing the data within records, you can use File Manager to edit the records within a data set.

The tasks described in this section are:

Note: Many of these actions are performed using prefix commands. Generally, you need to type over only the first 1 or 2 characters of the line number to enter a prefix command. Sometimes, however, typing a single character can be ambiguous. In the following example, it is unclear whether the intended prefix command is R to repeat line 31700, or R3 to repeat the line three times, R31 to repeat the line 31 times or R317 to repeat the line 317 times:

```
031600
R31700
031800
```

In such cases, the editor assumes that you have not typed a number following the prefix command. If you want to repeat the line three times, you can type one or more blanks following the R3.

```
R3 700
```

Inserting records

When the display format is set to CHAR, HEX, LHEX or TABL formats, you can insert new records into the data set. You cannot insert new records when the display format is SNGL.

To insert new records in a data set:

1. Position your cursor in the prefix area of the record above which you want to insert new records.
2. Type the I or I*n* prefix command, where *n* represents the number of records to be inserted.
3. Press Enter. The new record or records are inserted above your cursor position. Each record is initialized as follows:
   - If the display format is CHAR, HEX or LHEX, then the entire record is initialized to blanks.
   - If the display format is TABL, then the record is initialized according to the record structure of the current record type:
     - Numeric fields are initialized to zero.
     - Alphanumeric fields are initialized to blanks.
     - If the record type contains a variable-length array, then the record is initialized with the minimum number of array elements, and the array size field (or fields, for multi-dimensional arrays) is initialized accordingly.

Note: The inserted records are not saved to the file until a SAVE, FILE, END or EXIT command is issued.

Deleting records

You can delete records when in any display format, by using the DELETE primary command. This allows you to delete all the records in a data set or to delete multiple records based upon the following common factors:

- All excluded or not-excluded records (see “Excluding records” on page 112).
- All records that fall within a range specified by labels (see “Labelling record ranges” on page 113).
- All excluded or not-excluded records that fall within a labeled range.

To delete records that have a common factor:

1. Establish the common factor for your records, that is, create your labels and/or exclude the required records.
2. Enter the DELETE primary command, together with your choice of parameters, on the Command line.

Examples

The following command deletes all excluded records:

```
DELETE ALL EX
```

The following command deletes the next not-excluded record in the range of records from the label .LABA to the label .LABB:

```
DELETE NX .LABA .LABB
```

In CHAR, HEX, LHEX or TABL display formats, you can also delete records by using the D prefix command. Using the prefix command allows you to delete individually selected records or blocks of records that have no other common factor.

To delete individually selected records:

1. Perform either of these actions:
   - Position your cursor in the prefix area of the first record to be deleted and type the D or D*n* prefix command, where *n* represents the number of records to be deleted.
   - Type DD in the prefix area of the first record of a block of records to be deleted and DD in the prefix area of the last record in the block.
2. Press Enter. The record or record are deleted from the data set.

Note: The deletion of these records is not saved to file until a SAVE, FILE, END or EXIT command is issued. You can recover these deleted records, before ending your editing session, by using the RECOVER primary command.

Recovering deleted records

Records that are deleted during an editor session are stored in a buffer. At any time before closing the editor panel, you can recover these deleted records.

To recover previously deleted records:

1. Enter `RECOVER` *n* on the Command line, where *n* represents the number of records to be restored, starting with the last deleted record.

Records are stored in the buffer in the order in which they were removed, so that the command `RECOVER 1` restores the last record to be deleted, `RECOVER 2` restores the last two records to be deleted, and so on. Recovered records are inserted after the first record visible at the top of the data area. If the first visible line is `****** **** Top of data ****`, the records are inserted at the beginning of the data set. Inserted records appear in the order in which they were deleted.

For example, if records 1, 5 and 7 were deleted from the data set in that order, the command `RECOVER 1` would restore record 7, but `RECOVER 3` would restore records 1, 5 and 7 in that order (not as 7, 5, 1).

Recovering a record removes it from the buffer. If you delete the recovered record, before saving the first recovery, you cannot recover it a second time. This can lead to loss of data. To prevent data loss, issue the SAVE command after each RECOVER command.

Note: When using an auxiliary edit, in order to minimize the recovery buffer usage and allow for many records to be deleted, only the ten latest deleted records are kept in the recovery buffer.

Moving records

You can rearrange the display of your records by moving them up or down in the data set. For record types other than KSDS, the new sequence of the records is stored when you save the data set. In a KSDS, moving records while editing changes the displayed position of records, but has no effect on the stored data set. For this reason, the Move prefix commands as described in the following are not available for a KSDS. If you want to change the order of records in a KSDS, you must change the key values.

To move records within a data set:

1. Position your cursor in the prefix area of the first record that you wish to move.
2. Type the M or M*n* prefix command, where *n* represents the number of records to be moved, or type MM on the first of a block of records to be moved and MM in the prefix area of the last record in the block.
3. Press Enter. The prefix command is highlighted and the message "Prefix command pending" is displayed in the upper right corner of your screen.
4. Move your cursor to the prefix area of the record where you would like to place the moved records.
5. Type one of the following prefix commands:

   A This command inserts the moved records *after* the current location.

   B This command inserts the moved records *before* the current location.
6. Press Enter. The records are removed from their previous location and inserted at the new position.

Note: When you move records with the M, M*n*, or MM prefix commands, instead of moving the data after or before other records, you can choose to *overlay* the moved records on existing records in the data set you are editing. For details, see “Overlaying data in existing records” on page 119.

Copying and repeating records

You can create copies of records in two different ways: by copying or by repeating records. Copying allows you to place the copied records anywhere in the data set. Repeating allows you to quickly duplicate a record, immediately after the selected record.

To copy records within the data set:

1. Position your cursor in the prefix area of the first record that you wish to copy.
2. Type the C or C*n* prefix command, where *n* represents the number of records to be copied, or type CC on the first of a block of records to be copied and CC in the prefix area of the last record in the block.
3. Press Enter. The prefix command is highlighted and the message "Prefix command pending" is displayed in the upper right corner of your screen.
4. Move your cursor to the prefix area of the record where you would like to place the copied records
5. Type one of the following prefix commands:

   A This command inserts the copied records *after* the current location.

   B This command inserts the copied records *before* the current location.
6. Press Enter. The copied records are inserted at the new position.

Note: When you copy records with the C, C*n*, or CC prefix commands, instead of copying the data after or before other records, you can choose to *overlay* the copied records on existing records in the data set you are editing. For details, see "Overlaying data in existing records."

To repeat records:

1. Position your cursor in the prefix area of the first record that you wish to repeat.
2. Type the R or R*n* prefix command, where *n* represents the number of records to be repeated, or type RR on the first of a block of records to be repeated and RR in the prefix area of the last record in the block.
3. Press Enter. The copied records are inserted immediately below the last record selected for repeating.

Overlaying data in existing records

When data is to be copied or moved by the C (copy) or M (move) prefix commands and overlaid on one or more existing records, the O (overlay) prefix command specifies the destination for the data.

The data that is copied or moved overlays blanks in the destination records. This allows you to rearrange a single-column list of items into multiple column, or tabular, format.

When data is to be moved or copied and then overlaid on a destination:

- Where the destination is a *single record*:
  - The O (overlay) prefix command specifies the destination for the data.

    You can type a number after the O prefix command to specify the number of times that the M or C prefix command is to be performed. For example, typing the command `O3` against a record causes the data to be moved or copied and then overlaid on that record and also the next two records.
- Where the destination is a *block of records*:
  - The OO (overlay, multiple-line target) prefix command specifies the first and last record of the destination for the data.

To overlay one or more *single* records:

1. Type either `M` or `C` in the prefix area of the record that is to be moved or copied.
2. Type `O` in the prefix area of the record that the moved or copied record is to overlay.
3. Press Enter. The data being moved or copied overlays the specified record or records.

To overlay one or more *blocks* of records:

1. Type either `MM` or `CC` in the prefix area of the first and last records of a block of records that is to be moved or copied. You can scroll (or use FIND or LOCATE) between typing the first command and the second command, if necessary.
2. Type `OO` in the prefix area of the first and last records that the block of records being moved or copied is to overlay. Again, you can scroll (or use FIND or LOCATE) between typing the first `OO` and the second `OO`, if necessary.
3. Press Enter. The records that contain the two CC or MM commands and all of the records between them overlay the records that contain the two OO commands and all of the records between them.

Only blank characters in the records specified with O or OO are overlaid with characters in the corresponding columns from the source records. Characters that are not blank are not overlaid. The overlap affects only those characters within the current column boundaries.

The number of source and receiving records need not be the same. If there are more receiving records, the source records are repeated until the receiving records are gone. If there are more source records than receiving records, the extra source records are ignored. The overlay operation involves only data records. Special lines such as BNDS and COLS are ignored as either source or receiving lines.

Figure 49 illustrates the O (overlay) prefix command. Suppose you were editing a list in a single left-adjusted column and wanted to place portions of the list side-by-side. First, using the) (column shift right) command, shift a portion of the list the appropriate amount to the right to overlay in a multiple column format. Next, type MM in the prefix area to mark the beginning and end of the block of records you want to move, then type OO in the prefix area to mark the destination of the records you want to overlay.

```
 Process   Options   Help
_____________________________________________________________________________
Edit             JOHNLEV.TEST.DATA(OLAY)                          Top of 16
       Col 1_____ Insert Length 80_____         Record AT TOP___ Format CHAR
       ----+----10---+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
000002
mm0003 THESE RECORDS TO BE COPIED
000004 THESE RECORDS TO BE COPIED
000005 THESE RECORDS TO BE COPIED
000006 THESE RECORDS TO BE COPIED
mm0007 THESE RECORDS TO BE COPIED
000008
oo0009                               THESE RECORDS ARE THE TARGET
000010                               THESE RECORDS ARE THE TARGET
000011                               THESE RECORDS ARE THE TARGET
000012                               THESE RECORDS ARE THE TARGET
000013                               THESE RECORDS ARE THE TARGET
oo0014                               THESE RECORDS ARE THE TARGET
000015 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
Command ===> _____________________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 49. Before the O (Overlay) prefix command

When you press Enter, the editor overlays the records you marked to move on the destination block. See Figure 50 on page 121.

```
 Process   Options   Help
_____________________________________________________________________________
Edit             JOHNLEV.TEST.DATA(OLAY)                          Top of 16
       Col 1_____ Insert Length 80_____          Record AT TOP___ Format CHAR
       ----+----10---+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
000002
000003
000004 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000005 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000006 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000007 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000008 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000009 THESE RECORDS TO BE COPIED    THESE LINES ARE THE TARGET
000010 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
000011
****** ****  End of data  ****



Command ===> _____________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 50. After the O (Overlay) prefix command

Related topics

"Editor panel" on page 534
"View panel" on page 707

Copying data to and from a clipboard

In edit, the CUT and PASTE primary commands allow you to copy data to and copy data from a clipboard respectively.

Copying data to a clipboard (CUT)

Use the CUT primary command to move or copy one or more records from the current edit session into a clipboard for later retrieval by the PASTE command.

To specify the records to be put into the clipboard, use:

- Prefix commands C or CC to copy records.
- Prefix commands M or MM to move (copy, then delete) records.
- A range of line labels to copy records. For example:

```
CUT .ZF .ZL
```

If you do not specify a range, all edit session data is copied to the clipboard.

You can use the X or NX operands to copy only excluded or non-excluded data within the range to the clipboard. The default is both excluded and non-excluded data.

Suppressed and not-selected sets are not copied to the clipboard. To include the data referenced by those sets, first expose the sets by issuing the appropriate SHOW command.

To *replace* the contents of the clipboard, use the REPLACE operand. This is the default.

To *add to* the existing contents of the clipboard, use the APPEND operand.

To copy only excluded or unexcluded records to the clipboard, use the X or NX operand.

If you do not specify a clipboard name on the command, File Manager saves the data to the clipboard, DEFAULT.

When running under ISPF, clipboards created by File Manager can be accessed by ISPF. Similarly, when running File Manager, clipboards created when running under ISPF can be accessed by File Manager.

To view or edit an existing clipboard, issue the command `CUT DISPLAY` to display the Clipboard Manager panel. This panel lists all the current clipboards.

Copying data from a clipboard (PASTE)

Use the PASTE primary command to copy records from a clipboard into the current edit or view session.

If records in the clipboard are longer than the record length of the data set being edited, the records are truncated.

Use the A or B prefix commands, or AFTER or BEFORE with a label, to specify the destination of the records.

If you specified a name of a clipboard with the CUT command, you must use that name on the PASTE command to retrieve the contents of that clipboard.

To empty the clipboard after the PASTE completes, use the DELETE keyword.

To copy the records from the clipboard (instead of moving them), use the KEEP keyword.

Related topics

"CUT primary command" on page 756
"PASTE primary command" on page 797
"Clipboard Manager panel" on page 468

Copying data from another data set

The COPY primary command allows you to copy one or more lines of data from a data set or member of a partitioned data set (PDS) into the member or data set you are currently editing or viewing.

Use the A (after) or B (before) line commands, or the AFTER or BEFORE keyword with a label, to specify where the data is to be copied.

If you specify a member name as part of the COPY primary command, and you have properly specified the "after" or "before" destination, the entire member or a range of records in the member are copied in immediately.

Examples:

COPY *memnam*
: Copies in entire contents of member *memnam*.

COPY (*memnam***) 2 10**
: Copies in lines 2 to 10 of member *memnam*.

You can specify a partially qualified or fully qualified data set name as part of the command. This can be a data set or another partitioned data set.

Examples:

COPY *seqds*
Copies in entire contents of sequential data set *seqds*.

COPY '*userid.seqds*'
Copies in entire contents of sequential data set *userid.seqds*.

COPY *pds*(*mem*)
Copies in entire contents of member *mem* within partitioned data set *pds*.

COPY *pds*(*mem*) **6 10**
Member list is displayed. Copies in lines 6 to 10 of member *mem* within partitioned data set *pds*.

If you enter the COPY command without specifying a member name or a data set name, after you have specified a destination and pressed Enter, File Manager displays the Edit/View - Copy panel.

Example:

COPY AFTER .HERE
Displays the Edit/View - Copy panel.

The Edit/View - Copy panel allows you to enter the name of a data set or PDS (and member) containing the data you want to copy. If you do not want to copy the entire contents, you can specify the numbers of the first and last records to be copied.

Related topics
"COPY primary command" on page 755
"Edit/View - Copy panel" on page 544

Splitting and joining lines

At times, you might want to split a record in two, or join two records together. This is particularly appropriate when editing data sets that contain unstructured text (such as a REXX procedure), as in these data sets, a record represents a line of text.

Note: You cannot split or join a record when in SNGL or TABL display format, or when zoomed in on a record.

To split a record in two:

1. Type the SPLT primary command on the Command line.
2. Position your cursor on the record that you want to split, at the column position that you want to "push" to the next line.
3. Press Enter. A new record is inserted below your current record and the text from your cursor position to the end of the record is moved to the beginning of this record.

For example, when the following record is split at the cursor position (underlined character):

```
****** ****  Top of data  ****
000001 abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
000002 abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
```

the result is:

```
****** ****  Top of data  ****
000001 abcdefghijklmnopqrstuvwxyz
000002 abcdefghijklmnopqrstuvwxyz
000003 abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
```

When you join two records, the text from the second record is combined with the first record, overlaying the first record starting at the cursor position.

To join two records together:

1. Type the JOIN primary command on the Command line.
2. Position your cursor on the record that you want to combine with the next record, at the column position where you want the second record text to begin.
3. Press Enter. The text from the following record is combined with the current record, overlaying any existing text.

   If the text in the second record does not fit into the remaining space in the first record, it spills over into a new record.

For example, when the following records (length=50) are joined at the cursor position (underlined character):

```
       ----+----1----+----2----+----3----+----4----+----5
000000 ****  Top of data  ****
000001 aaaaaaaaaaaaaaaaaaaaaaaaa
000002 bbbbbbbbbbbbbbbbbbbbbbbbb
000003 ccccccccccccccccccccccccccccccccccccccccccccccccc
000004 ****  End of data  ****
```

the result is:

```
       ----+----1----+----2----+----3----+----4----+----5
000000 ****  Top of data  ****
000001 aaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbb
000002 ccccccccccccccccccccccccccccccccccccccccccccccccc
000003 ****  End of data  ****
```

However, if these records were joined at the same position:

```
       ----+----1----+----2----+----3----+----4----+----5
000000 ****  Top of data  ****
000001 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
000002 bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
000003 ccccccccccccccccccccccccccccccccccccccccccccccccc
000004 ****  End of data  ****
```

the result would be:

```
       ----+----1----+----2----+----3----+----4----+----5
000000 ****  Top of data  ****
000001 aaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbb
000002 bbbbbbbbbbbbbbbbbbbbbbbbb
000003 ccccccccccccccccccccccccccccccccccccccccccccccccc
000004 ****  End of data  ****
```

To avoid the danger of unintentionally overlaying data in the first record, you can use the SPLTJOIN command, which can be abbreviated to SJ. This command splits

or joins records depending upon the cursor position within a record. If the cursor is after the last non-blank character in a record, the command performs a join. If not, it splits the record.

Note: You may want to assign the SJ command to a function key, using the KEYS command (see the *ISPF User Guide Volume 1* for information on how to assign commands to function keys).

Changing the length of a record

For those files that support variable-length records, you can change the length of the record by displaying the record length in an edit session and then overtyping it with the required value.

Note: You cannot change the length of a record in In-place edit.

You can display the record length by any of these methods:

- Entering the zoom mode from a CHAR, HEX, or LHEX display format.
- Displaying the record in SNGL display format.
- In multi-line formats (TABL, CHAR, HEX, and LHEX), issuing the RECLEN primary command (`RECLEN ON`, `RECLEN LEFT`, or `RECLEN RIGHT`).
- Ensuring the **Display record length** option on the relevant Editor Options panel is selected.

Note: If you increase the record length, the new bytes added to the record are blanks.

Related topics

"Displaying the record length" on page 74
"RECLEN primary command" on page 803
"Zooming in to see all of a record" on page 71
"Editor Options panel" on page 546
"In-place edit" on page 98

Changing the length of a record

Chapter 4. Creating and editing templates

For many tasks performed with File Manager, you can manipulate your view of a data set by applying a template. Templates can be pre-built for later use or prepared "on the fly" within File Manager panels. You can also use templates when programming File Manager functions in batch jobs, REXX procedures and TSO CLISTS (described in Chapter 12, “Introduction to programming with File Manager functions,” on page 397 and Chapter 13, “Enhancing File Manager processing,” on page 405).

The major tasks and concepts described in this chapter are:

Template types and structure

A File Manager template is a collection of information that you can use to select and format records and fields in an application data set. The template provides File Manager with a logical view of the data to be used when editing, viewing, copying, printing and creating data sets. Templates can be based upon the definitions in COBOL copybooks, PL/I INCLUDE, and HLASM copy members. Additionally, record structure descriptions can be extracted from complete COBOL, PL/I, or HLASM source programs. Alternatively, you can create your own record structure description in a dynamic template.

The concepts and tasks described in this section are:

Information stored in a template

The list below describes the information that a template can contain, and where and how File Manager uses the information.

Record type selection
Determines which record types are selected or unselected.

Unselected record types are excluded when comparing, copying or printing.

When creating a data set, the records in the new data set are of the record type first selected in the template.

Record identification criteria
Determines which records belong to each record type in the template.

Records that do not belong to a record type in the template are indicated as “not selected” when viewing or editing, and are excluded when comparing, copying[1] or printing.

Record selection criteria
Determines which records, having been identified as belonging to a record type, are to be selected.

Records that do not match the record selection criteria are indicated as “not selected” when viewing or editing, and are excluded when comparing, copying[1] or printing.

Records that are not selected may not be included in the editor session, depending on the setting of the **Include only selected records** option. Furthermore, if the template describes segmented data, the record selection criteria affects the selection of the entire physical record. When this occurs, none of the segments belonging to the physical record are provided in an editor session and they are excluded when comparing, copying, or printing.

Field selection
Determines which fields are displayed (when viewing or editing) or printed. This information is ignored when comparing, copying or creating data sets.

When copying, if you want to exclude some fields in the input data set from being copied to the output data set, then you need to create a “To” copybook or dynamic template without those fields.

When comparing, if you want to exclude fields from the comparison, you need to create an “Old” and a “New” copybook or dynamic template and then remove the default field mapping for the fields to be excluded.

Field sequence
Overrides the default order in which fields are displayed (when viewing or editing) or printed. By default, fields are displayed or printed in the order in which they were defined in the copybook. This information is ignored when comparing, copying or creating data sets.

When copying, if you want to change the order of fields in the input data set and the output data set, then you need to create a “To” copybook or dynamic template with the fields in the desired order.

Key sequence
Determines the sequence of fields to be included in a multi-segment key, which can be used when performing a file comparison with keyed synchronization.

Field headings
Overrides the default field headings when viewing, editing, comparing or printing. (The default field headings are the field names defined in the copybook.)

Field attributes
The field name, data type, starting column and length, as defined in the dynamic template or in the copybook on which a template is based. These attributes are fixed in a copybook template, as they are determined by the underlying copybooks, but can be edited in a dynamic template.

If you make minor changes to field attributes in a copybook, you can (with some restrictions) update the template generated from it, while keeping all of the other existing information in the template.

Field use attributes
Consist of two attributes:

Output width
Defines the number of columns allocated to the field when viewing, editing or printing. By default, this is the width defined in the copybook by the picture specification.

If a numeric field contains a valid numeric value, but the value is too large to fit in the output width, then, when viewing or editing in SNGL or TABL display format, the value is truncated and highlighted.

Leading zeros (numeric fields only)
Defines whether or not the field value is shown with leading zeros when viewing, editing, comparing or printing.

Field create attributes
Defines the value to which the field is initialized when copying[2] or creating a data set.

Field mapping
Defines which fields in the "From" template map to fields in the "To" template, when copying[2] a data set; or which fields in the "Old" template map to fields in the "New" template, when comparing two data sets.

Scrambling options
Determine how the contents of the field are scrambled (if at all) during a copy process.

Note:

1. When copying, File Manager uses this information in the "From" template. In a "To" template, this information is ignored. However, when comparing, the selection information in both the "Old" and "New" templates (if specified) is used.
2. When copying, File Manager uses this information in the "To" template. In a "From" template, this information is ignored.

Related topics
"Copying data sets" on page 253
"Comparing data sets using templates" on page 284
"Managing templates with the Template Workbench" on page 167

About copybook templates

Copybook templates are templates where the *field definitions* are derived from one or more copybooks. A copybook is a member containing either COBOL data descriptions, PL/I DECLARE statements, or HLASM data definitions. Every copybook template has a *source definition*.

Source definition
The source definition describes the copybook members, their corresponding data sets, and how the field definitions are to be arranged in a template. The simplest form of source definition is a *single* copybook that contains all the record layouts required in their correct form. Single copybook source definitions can be referred to directly and a copybook template is built automatically from them.

Alternatively, you can provide an *advanced copybook source definition* that describes one or more copybook members, inserted level-01 data items, the

range of statements to be included, and how REDEFINES or UNION clauses are to be interpreted in terms of record layouts.

A single copybook source definition can be the entire source of a program or only field definitions.

An advanced copybook source definition must refer to fields definitions for the same language, and the source must be members of a partitioned data set.

The copybook template combines the layout information provided by the source definition with additional information supplied by the user pertaining to formatting, reformatting, record selection and data creation, to produce a logical view of the data on which these functions can be performed.

In a copybook template, the field definitions (start position, length and type) are not editable. However, copybook templates allow you to define a number of different formats for use with data sets that contain multiple record types. Records can be identified by type within a data set and each type can be associated with a different format. This allows you to view data with different types at the same time, and to view, edit or copy these records simultaneously. Copybook templates also allow criteria editing by field or by a free format REXX selection expression.

To create a new copybook template, you begin with one or more copybooks containing field definitions that describe the record structure of your data. When you specify the copybook (or copybooks) on a panel, File Manager compiles the descriptions into a template that you can save and reuse with any application data set that has the same record structure.

In a COBOL copybook, each level-01 group item describes a record type in the application data set. The elementary items in the group describe fields in the record type. For example, the following copybook describes the record structure of an application data set that contains two record types, ORDERS and ITEM:

```
01 ORDERS.
   05 ORDER-ID        PIC X(5).
   05 CUSTOMER-ID     PIC X(5).
   05 ORDER-DATE.
      10 ORDER-YEAR   PIC 9(4).
      10 ORDER-MONTH  PIC 9(2).
      10 ORDER-DAY    PIC 9(2).
01 ITEM.
   05 PRODUCT-ID      PIC X(9).
   05 QUANTITY        PIC 9(4) BINARY.
   05 UNIT-COST       PIC 9(8) BINARY.
```

The ORDERS records contain five fields: ORDER-ID, CUSTOMER-ID, ORDER-YEAR, ORDER-MONTH and ORDER-DAY. The ITEM records contain three fields: PRODUCT-ID, QUANTITY and UNIT-COST.

PL/I copybooks are similar to COBOL copybooks, in that the major structures (level-1 names) describe record types and elementary names describe fields:

```
DECLARE 1 ORDER,
          2 ORDERID       CHAR(9),
          2 CUSTOMERID    CHAR(5),
          2 ORDERDATE,
            3 ORDERYEAR   CHAR(4),
            3 ORDERMONTH  CHAR(2),
            3 ORDERDAY    CHAR(2);
DECLARE 1 ITEM,
          2 PRODUCTID     CHAR(9),
          2 QUANTITY      BIN(15),
          2 UNITCOST      BIN(31);
```

HLASM copybooks are similar to COBOL copybooks, in that the major structures (DSECT names) describe record types and elementary names describe fields:

```
ORDER      DESCT
ORDERID    DS  CL9
CUSTOMERID DS  CL5
ORDERDATE  DS  0CL8
ORDERYEAR  DS  CL4
ORDERMONTH DS  CL2
ORDERDAY   DS  CL2
ITEM       DESCT
PRODUCTID  DS  CL9
QUANTITY   DS  H
UNITCOST   DS  F
```

For information about coding copybooks, see the *IBM COBOL Language Reference*, *IBM VisualAge PL/I Language Reference*, or the *HLASM V1R5 Language Reference*.

Related topics

"Specifying or updating the source definition for a copybook template" on page 146
"Advanced copybook selection" on page 147
"Set Language and Compiler Specifications (option 0.5)" on page 654
"Compiler Language Selection panel" on page 470

Segmented data templates

File Manager offers template support for segmented records. A segmented record is a record that contains one or more logical segments, each of which is defined and identified by a copybook layout. When you apply a "segmented data template" to these records, each segment is treated as a logical record within the supported functions.

Figure 51 on page 132 shows a simple example of segmented records, seen without using a template:

```
----+----1----+----2----+----3----+----4----+----5----+
****  Top of data  ****
01FINAccountant          020654John      Browning  1875
01FINPurchasing Officer  021759Frederick Smith     1893
01FINAccounts Receivable 024163Annette   Fielding  1856
01MKTSales Representative026584Jessica   Simpson   1862
01MKTPromotions Manager  023579Catherine Jones     1812
01MKTMarket Research     021239Alan      Johnson   1814
01ADMChief Executive     020124David     Arrowsmith1801
01ADMSecretary/PA        024781Maryanne  Davies    1802
01ADMReceptionist        026329Lizette   Wilson    1800
01ADMClerical Officer    023571Stephen   Hughes    1806
****  End of data  ****
```

Figure 51. Segmented records with no template

Figure 52 shows the copybook definitions needed to define each record segment:

```
----+----1----+----2----+----3----+----4----+----5
****  Top of data  ****
       01 DEPT.
          03 Rec-Type     PIC XX.
          03 Dept-ID      PIC XXX.
          03 Job-Desc     PIC X(20).
       01 EMPLOYEE.
          03 Rec-Type     PIC XX.
          03 Emp-ID       PIC 9999.
          03 Given-Name   PIC X(10).
          03 Family-Name  PIC X(10).
          03 Extension    PIC 9999.
****  End of data  ****
```

Figure 52. Copybook definition for segmented records

Figure 53 on page 133 shows the DEPT. records in TABL format, after a segmented data template (generated from the copybook) has been applied.

```
REC-TYPE DEPT-ID JOB-DESC
#2       #3      #4
AN 1:2   AN 3:3  AN 6:20
<>       <->     <---+----1----+---->
01       FIN     Accountant
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       FIN     Purchasing Officer
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       FIN     Accounts Receivable
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       MKT     Sales Representative
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       MKT     Promotions Manager
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       MKT     Market Research
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       ADM     Chief Executive
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       ADM     Secretary/PA
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       ADM     Receptionist
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
01       ADM     Clerical Officer
- - - - - - EMPLOYEE  - - - - - - - - - - - 1 Line(s) suppressed
****  End of data  ****
```

Figure 53. TABL view of segmented data with the DEPT Record Type selected.

The following utilities/functions can support the use of a segmented record template:

View and Edit, DSU and DSEB functions
When a segmented data template is in use, File Manager uses In-place Editing. Segments cannot be inserted, deleted, copied or moved. The contents of segments can be edited but the length cannot be changed.

The BOUNDS primary command and BND prefix command cannot be used. If you really want to do a change across some common portion of all segments (or across certain columns of a certain segment), you can use the column range parameters on the FIND or CHANGE commands.

The Audit Report identifies changed segments.

Copy utility and DSC function
When copying a data set, you can only apply a segmented data template to the input data set. The output data set will contain a separate record for each segment in the input data set.

Compare utility and DSM function
Segmented data templates can be used for either the “Old” or “New” data sets or both. This allows you to compare segmented and non-segmented data. For example, after copying segmented data to separate records in a new data set, you might want to compare the data sets to ensure that no errors occurred in the copying process.

Print utility and DSP function
Segmented data templates are supported in the Print Utility and DSP function.

Note: Segmented data templates cannot be used for ISPF Packed data sets.

Related topics

"Setting up a template to be used with segmented data" on page 151
"Viewing segmented data" on page 229
"Copying records with segmented data templates" on page 262

Mapping SMF data with segmented templates

SMF records typically contain a header section with fields describing the following sections.

Typically three fields are used to tell you the offset, length and number of occurrences of a section. File Manager provides three functions that can be used in related identification criteria to aid in mapping these with a segmented template. The three segmented functions are SEGOFF(), SEGLEN() and SEGCNT().

A sample XML template illustrating the usage of these functions in mapping the SMF type 30 record can be found in your sample library (hlq.SFMNSAM1) in member FMNSMF3X. This sample XML refers to the assembler source member FMNSMF3S also found in your sample library. To use the sample XML template copy it and modify the `<library>hlq.SFMNSAM1</library>` to reflect the correct library data set for the source member at your site.

About dynamic templates

Dynamic templates are created by users and contain customized field definitions, consisting of the starting column, length and type of fields in the data set and, optionally, a field name to identify the information. These field definitions can be edited at any time. Dynamic templates are particularly useful when working with data sets that do not have an appropriate copybook.

A dynamic template can only contain one record layout. All fields defined are treated as Level 2 elementary items, with File Manager generating the Level 1 group item for the record. The generated Level 1 has a calculated record length, determined by the rightmost byte referenced by the field definitions. Record selection must be specified by field - you cannot specify a free format selection expression for a dynamic template.

Related topics

"Creating dynamic templates" on page 152

Data description support

Support for COBOL data description entries

In the COBOL data description entries specified in a copybook, File Manager supports:

- ASSIGN clause
- OCCURS clause (including OCCURS DEPENDING ON)
- PICTURE clause
- REDEFINES clause
- RENAMES clause
- USAGE clause
- VALUE clause

File Manager ignores:

- BLANK WHEN ZERO clause
- DATE FORMAT clause
- EXTERNAL clause
- GLOBAL clause

- JUSTIFIED clause

Note: In general, File Manager treats Numeric Edited fields as alphanumeric. A special code, ZA, is used for unsupported zoned decimals.

File Manager supports all data item level numbers except 88 (condition-names).

The literal value that you specify in a VALUE clause is used in the template to specify the **Start value** (for numeric fields) or **Pattern** (for alphanumeric fields) "create attribute" on the Field Attributes panel for that field.

File Manager uses this attribute to initialize the value of the field when creating data or inserting new fields when copying data.

Support for PL/I data description entries

In the PL/I data description entries specified in a copybook, File Manager supports:

- PL/I data attributes VARYING and VARYINGZ for single and double-byte character strings.
- PL/I bit variables including the VARYING attribute.
- Binary, decimal, floating point and numeric picture data (excluding symbols).
- REFER variables for array extents. Both upper and lower bound array elements may be declared for each dimension. The number of extents for each dimension is always:

  ```
  upper bound - lower bound + 1
  ```

  Note: A zero extent is not allowed for a PL/I dimension. If a bound is not specified PL/I assumes a value of 1.
- REFER variable to declare the length of a string variable, character (AN), varying-length character (VC) or zero-terminated character string (ZC) in bytes.
- File Manager also supports PL/I double-byte character variables (DB), varying-length double-byte character (VD) and zero-terminated double-byte characters (Z2) as byte pairs. File Manager also supports bit strings (BT) and varying bit string (VB) variables.
- REFER variables can be declared as integer, fixed binary, packed decimal, floating point, or character string variables.
- UNION declarations at any level which can have varying-length (REFER string length or REFER array extents).

  Note: UNIONS can contain overlays of different data types, which may require template workbench record selection criteria or SHOW control if data is not displayable.

File Manager ignores:

- INIT data values
- REFER start expression, for example, VAR(, expr Refer abc,) or CHAR(expr Refer abc).
- EXTERNAL attributes
- DEFINED structure overlays
- LIKE structure attributes

Note: In general, File Manager treats Numeric Edited fields as alphanumeric. A special code, ZA, is used for unsupported zoned decimals.

Support for variable-length arrays

A variable-length array is defined in a COBOL copybook by an OCCURS DEPENDING ON (ODO) clause, and in a PL/I copybook by a dimension attribute where the lower or upper bounds (or both) are defined by REFER options, or when a PL/I string data type has a REFER variable for the declared length, for example, CHAR(expr REFER name).

The following considerations apply when using a template that contains variable-length arrays:

- A record structure can contain multiple variable-length arrays; however, the fields that define the size of the arrays (the ODO or REFER "objects") must all appear in the record prior to the first variable-length array item, or, in the case of PL/I REFER length, prior to the first string declared with a REFER length.
- If any of the following are true:
  - (COBOL copybook only.) The object is out of the range specified by the ODO clause.
  - The actual record length does not match the calculated length of a record with the number of array items specified by the object.
  - (PL/I copybook only.) The record length does not match the calculated length of a record based on both string length REFERs and number of dimension extents calculated from the lower and upper bound REFERS for each dimension of an array.

    When editing the data set, the prefix area contains the description =LGTH, indicating that the record was not selected because of invalid length. (To show these records, rather than hiding them with other "not selected" records, select the **Length error** option on the Edit/Browse Options (option 0.8) panel.)
- In SNGL or TABL display format:
  - You can change the length of a record by changing the value of an object field. This is an exception to the rule that the length of a record cannot be altered in an Edit session, without using ZOOM.
  - When you change the value of an object field, then the number of items in the array expands or contracts automatically.

    New items are inserted at the end of the current array (and before any fields that follow the array); numeric items are initialized to zero and alphanumeric items are initialized to blanks.

    Similarly, items are deleted from the end of the array, and any fields subsequent to the end of the array are not affected.

    For PL/I, the number of items in an array is determined by subtracting the lower bound of the dimension attribute from the upper bound and then adding one. If any REFER variables (fields) are specified as a lower or upper bound of an array dimension, then changing the value of any REFER object field inserts or deletes array items accordingly. For the TABL Format, if the new number of array items exceeds the current number of TABL columns for the array, File Manager expands the number of columns in the display and adds new column headings. Similarly, if a PL/I string length REFER is changed, the column width in TABL for each variable using that REFER is changed.
  - (COBOL copybook only.) File Manager does not allow you to change the object field to a value outside the range specified by the ODO clause (File Manager supports an ODO lower value of 0.)

  - (COBOL copybook only.) Inserting a new record results in a record with the minimum number of array items specified by the ODO definition. For example, given the following ODO clause:

    ```
    ODO-ONE OCCURS 0 TO 10 TIMES DEPENDING ON ODO-CNTRL-ONE.
    ```

    then, when you insert a new record, the ODO-CNTRL-ONE field is initialized to zero, and there are no ODO-ONE array items in the new record.

    Given the following ODO clause:

    ```
    ODO-ONE OCCURS 5 TO 7 TIMES DEPENDING ON ODO-CNTRL-ONE.
    ```

    then ODO-CNTRL-ONE is initialized to 5, and there are five ODO-ONE array items in the new record.
  - (PL/I copybook only.) Inserting a new record using a template that contains REFER specifications results in one element for each array dimension. Upper and lower bound REFER values are set to one.
- In CHAR, HEX or LHEX display format, changing the value of an object field does **not** change the record length. In these display formats, changing the value of an object field can result in a mismatch between the object field value and the calculated length of the record with that many array items.

  If you select **Expose (do not group) records of types: Length error** on the Edit/Browse Options (option 0.8) panel then, in Edit (option 2), such records appear with an =LGTH indicator in their prefix area.

Editing rules for PL/I varying length string types

Fields with the PL/I VARYING attribute (VC, VD, and VB) are separated into two fields for File Manager processing. The first field is the current string length field (named "len" in File Manager Headings), and is followed by a second field that contains the string data to be displayed. Each of the string types has the current string length displayed as a FIXED BIN(15) field.

The separated “len” fields created by File Manager for varying strings do not need creation attributes, as the data content determines the “len” value.

In Edit mode, the data string field maximum display width is divided into unprotected characters, up to and including the current length, and a protected area beyond the current length in a non-display mode. VD strings require two character positions for each double-byte value declared for the string, and two character positions for each current length value that is unprotected. BIT strings require one character position in the Edit panel for each bit value in the string. Character positions beyond the current string length are in protected non-display mode for VD and VB types.

The maximum string length for these string types can be declared as a constant or as varying by specifying a REFER variable name that is present in the structure. If declared with a REFER variable for maximum string length, then each record can have different starting positions and data string display widths for that field and any following fields. REFER variables may also be present in the structure to define the bounds or extents of variables declared as dimensioned arrays.

The Edit processing rules for a VARYING field are:

- The length field of a varying string can be changed to a value in the range 0 to the maximum declared length for the field in that record.
- If the length is decreased, then the data string is truncated to the new length and the remaining display positions are protected from entry.

- If the length is increased, appropriate initialization occurs up to the new current length.
- To change the maximum length of a string variable that uses a REFER value, that REFER value must be changed first and the ENTER key pressed to reset the maximum string length. If the new maximum set by the REFER value is less than the current length of the data string, the data string is truncated and the current length is highlighted as a field in error.

Editing rules for PL/I varying length zero terminated string types

Character or Graphic data types with the PL/I VARYINGZ attribute (FM types ZC and Z2) are shown in the Edit panels with one or two display characters added to the end of the declared length. These screen positions contain X'00' or X'0000' when the current length of the string is equal to the maximum declared length. The extra positions are protected from data entry by File Manager.

The Edit processing rules for a VARYINGZ field are:

- The position of the X'00' or X'0000' terminating characters is automatically moved within the string if the replacement string length is not equal to the replaced string length.
- If the updated string length would exceed the string maximum, an error message is issued.
- If the updated string length is equal to the string maximum the X'00' or X'0000' is moved to the extra display positions.
- If a VARYINGZ field declaration specifies a maximum length REFER variable, the displayed REFER value may be updated to increase or decrease the maximum string length for the VARYINGZ field. If the new maximum length set by the REFER value is less that the current length of the data string, the data string is truncated and the terminating characters are placed in the new extra positions.
- Overtyping of a Graphic VARYINGZ field is not recommended as Graphic strings usually contain undisplayable characters that can't be overtyped in the TABL or SNGL Formats.

Using HLASM copybooks with File Manager

To instruct File Manager to process a HLASM (High Level Assembler) copybook, you must set the language option to HLASM. The **Auto detect** selection applies to COBOL and PL/1 copybooks only. Once you have set the option to HLASM, then every function that supports copybooks can be used with a HLASM copybook. Use the HLASM compiler specifications to provide extra SYSLIB and change ALIGN and DBCS processing options.

Related topics

"Setting your HLASM processing options" on page 144

Interpreting HLASM source

To process a HLASM copybook, File Manager invokes the HLASM compiler. It then produces a record layout (or segment layout) for each DSECT it finds with corresponding DS or DC fields. File Manager interprets the DS and DC fields as shown here:

Blank field name
: If a DS or DC statement does not have a field name, then it is assigned the name "*".

Multiple operands

If a DS or DC statement has multiple operands, then the field isdefined as a group field and a child element is created for each operand with the suffix of "_O*n*" where *n* is the relative number of the operand.

For example:

```
REC_TYPE01 DSECT
FIELD1  DS    F,H,CL1
```

produces:

```
Ref Field Name                        Picture  Type Start    End Length
    ****  Top of data  ****
  1 1 REC_TYPE01                               AN      1      7      7
  2  2 FIELD1                           F,H,CL1  AN      1      7      7
  3   3 FIELD1_O1                               BI      1      4      4
  4   3 FIELD1_O2                               BI      5      6      2
  5   3 FIELD1_O3                               AN      7      7      1
    ****  End of data  ****
```

Multiple values

If a DS or DC operand has multiple values associated with it, then a group field is defined with child elements for each value with the suffix of "_V*n*" where *n* is the relative number of the value.

For example:

```
REC_TYPE01 DSECT
FIELD1  DS    F'1,2,3',H,CL1
```

produces:

```
Ref Field Name                   Picture       Type Start    End Length
    ****  Top of data  ****
  1 1 REC_TYPE01                               AN      1     15     15
  2  2 FIELD1                   F'1,2,3',H,CL1 AN      1     15     15
  3   3 FIELD1_O1                             AN      1     12     12
  4    4 FIELD1_O1_V1                         BI      1      4      4
  5    4 FIELD1_O1_V2                         BI      5      8      4
  6    4 FIELD1_O1_V3                         BI      9     12      4
  7   3 FIELD1_O2                             BI     13     14      2
  8   3 FIELD1_O3                             AN     15     15      1
    ****  End of data  ****
```

Duplication factor

If a DS or DC operand has a duplication factor greater than 1, then the resultant field is treated as a single dimensioned field that occurs the number of times specified by the duplication factor.

For example:

```
REC_TYPE01 DSECT
FIELD1  DS    5F
```

produces:

```
Ref Field Name                          Picture  Type Start    End Length
    ****  Top of data  ****
  1 1 REC_TYPE01                                  AN      1     20     20
  2  2 FIELD1(5)                          F       BI      1      4      4
    ****  End of data  ****
```

Bit length specification

If a DS or a DC has a bit length specification, it is defined as a bit field irrespective of the field type.

For example:

```
REC_TYPE01 DSECT
FIELD1  DS    AL.5
```

produces:

```
Ref Field Name                                Picture  Type Start    End Length
    ****  Top of data  ****
  1 1 REC_TYPE01                                       AN      1      1      1
  2  2 FIELD1                                 AL.5      BT      1      1      1
    ****  End of data  ****
```

DSECT or layout length
The level-1 field is always set to the maximum end column for any field.

Redefines/Range specifications
If you are using advanced copybook selection to build a template, the X prefix command enables you to interpret the ORG fieldname in the same manner as a COBOL redefines or PL/I union statement.

Managing templates

You can create or edit a copybook template or a dynamic template "on the fly", from any of the following panels:

- View Entry
- Edit Entry
- Utilities/Data Create
- Utilities/Print
- Utilities/Copy (From and To panels)
- Utilities/Compare (Old and New panels)

You can also create, edit and otherwise manage your templates, independently of any particular task, within the Template Workbench.

The tasks described in this section are:

"Specifying or updating the source definition for a copybook template" on page 146
"Creating a copybook template with the Template Workbench" on page 150
"Advanced copybook selection" on page 147
"Setting up a template to be used with segmented data" on page 151
"Creating a template from a model template" on page 151
"Creating dynamic templates" on page 152
"Editing a template" on page 156
"Managing templates with the Template Workbench" on page 167

Setting your template processing options

File Manager goes through the compilation process whenever you create a template from a copybook, even if you have already successfully created other templates from the same copybook, and whether you are creating a temporary template while viewing data, or creating a permanent template using the Template Workbench.

You can customize the way that File Manager compiles templates by changing your template processing options. You are able to choose whether File Manager detects the programming language that you are using in your copybooks and then automatically selects the appropriate compiler, or whether you want to force File Manager to always use one type of compiler. You can also set the processing options for the COBOL, HLASM, and PL/I compilers.

If you are creating a new template from an existing *template* model (using the CM command on the Template Workbench (option 7) panel), File Manager does not perform a compilation.

When you specify a copybook in a panel, batch job, REXX procedure or other function, File Manager:

1. Checks whether the copybook is a complete source program , or contains only field definitions.

   If the copybook is not a complete program, then File Manager includes the copybook in a shell program.
2. Invokes the compiler to check the syntax of the copybook, and to produce an ADATA file.

   You can use the Compiler Language Selection panel to:
   - Specify which compiler File Manager is to invoke, or whether File Manager is to automatically detect whether the language is COBOL or PL/I, and invoke the appropriate compiler.
   - Override the compiler options stored in the template.

   You can specify the acceptable return code level for the compilation in the language processing options panels.
3. If the compilation completes without errors, then File Manager processes the information in the ADATA file to create a template.

   If the compilation completes with a return code of greater than the specified maximum:
   - If you are using File Manager under ISPF, then a pop-up menu is displayed. From the pop-up menu you can:
     - View the compilation listing using Print Browse (option 3.9).
     - Abort the template creation process.
     - Retry the compilation. Before you select this option, view the compilation listing and correct any errors in the copybook. While you are viewing the compilation listing, you can use the ISPF split screen facility to swap to another ISPF session and use your editor to correct the errors in the copybook.
     - Ignore the errors and proceed with creating the template. This option is only available if File Manager is able to create a template. Some compilation errors, such as warnings, might have no effect on the creation of the template. If you are unsure, you can view the compilation listing. If the compilation error is too severe, File Manager cannot create a template.
     - Try the compiler for the other language.
   - If you are using a File Manager function in a batch job, REXX procedure or TSO clist, then File Manager issues the error message “Copybook Compilation Errors Occurred RC=*nn*”. The batch job, REXX procedure or TSO clist continues as if you had not specified a copybook.

Related topics

“Compiler Language Selection panel” on page 470
“Selecting your compiler language”
“Overriding Compiler Options” on page 145

Selecting your compiler language

To select your compiler language:

- From the Primary Option Menu panel, select **0 Settings**, then **5 Compiler**, and **1 LANG**.
- Specify your choice by entering the option number:

1. **COBOL**
   Use the COBOL compiler, regardless of the language used in your copybook.
2. **PL/I** Use the PL/I compiler, regardless of the language used in your copybook.
3. **Auto detect**
   Note: Applies to COBOL and PL/I only. To instruct File Manager to use the HLASM compiler, yuo must select **4. HLASM**.

   Allow File Manager to detect whether the language used in you copybook is COBOL or PL/I and select the appropriate compiler.

   If the copybook does not contain any keywords that File Manager can use to determine the language, then File Manager uses the COBOL compiler.

   If the compiler creates ADATA and produces a return code of less than or equal to the Maximum Return Code specified in the compiler processing options, then File Manager creates the template without any prompting.

   If the compiler creates ADATA but produces a return code greater than the specified maximum, then File Manager displays a pop-up menu where you can choose from various options, including using the other compiler.

   If no ADATA was created, then File Manager uses the other compiler without prompting.

   If the other compiler creates ADATA but the return code is greater than the specified, then File Manager displays the pop-up menu.

   If the other compiler does not create ADATA, then File Manager retries the first compiler, and again displays the pop-up menu.
4. **HLASM**
   Use the HLASM compiler.

Related topics
"Data description support" on page 134
"Compiler Language Selection panel" on page 470

Setting your COBOL processing options

When your COBOL copybooks fit into any of the following scenarios, you must set your COBOL processing options so that File Manager can process them correctly.

- You have used COPY compiler-directing statements to include other members that do not belong to the same PDSs as the copybooks.
- Your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates.
- You have used DBCS characters in your copybooks.

Note: File Manager now supports specifying additional SYSLIBs and REPLACE text in the same copybook.

To set your COBOL processing options:

1. From the Primary Option Menu panel, select **0 Settings**, then **5 Compiler**, and **2 COBOL**.
   The COBOL Processing Options panel is displayed.

2. If you have used additional SYSLIBS, enter the names of the data sets in the **Additional SYSLIB data set** fields.

   Members included by COPY compiler-directing statements must be found in the SYSLIB concatenation. When you specify a copybook on a File Manager panel to create a new template (or update an existing template), the SYSLIB concatenation consists of the PDS of the copybook, plus up to ten additional PDSs that you can specify on this panel. These PDSs are searched in order (PDS of the copybook, followed by these additional PDSs, 1–10.)

   Note: CA-Panvalet libraries cannot be specified as additional SYSLIB sets. Further, if a CA-Panvalet Library is the main copybook library, then no additional SYSLIB data sets may be specified.
3. If you want to replace text in your copybooks, enter pseudo text character strings in the COBOL Replacing Options **From string** and **To string** fields.

   For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `==:==` as a "From" string and `=====` as the matching "To" string.
4. If your copybooks contain DBCS characters, select the DBCS compiler option. Otherwise, use NODBCS.
5. To set the maximum acceptable return code for a compile, enter a value in the **Maximum Return Code to be accepted from compiler** field.

   When using the online panels to generate a template from a copybook, if the compiler returns a warning/error code greater than the specified number, File Manager displays a pop-up menu in which you can make decisions about further processing.

   When generating a template during the processing of a batch function, if the compiler returns a warning/error code greater than the specified number, File Manager halts the running of the function.
6. To use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma" when compiling COBOL copybooks, select the option, **Decimal-point is comma**.

   When you select this option, the COBOL compiler exchanges the functions of the period and the comma in PICTURE character strings and in numeric literals.
7. To use the Arith(extend) COBOL compile option when compiling COBOL copybooks, select the option, **Arith(extend)**.
8. To retain the original case of field names as coded in the COBOL copybooks, select the option, **Mixed case field names**. This feature is only available if you are running with the File Manager COBOL compiler or a minimum compiler level of Enterprise COBOL V4R1.

   If you do not select this option, field names are stored in uppercase.
9. Supply any additional compiler options which will be added via the CBL statement.
10. Press the Exit function key (F3) to save the changes you have made and exit the panel.

Related topics

"Set COBOL Processing Options panel (option 0.5.2)" on page 649

For details on specifying "From" and "To" strings for the COBOL REPLACE directive, see the *IBM COBOL Language Reference*.

For details on the effect of these compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

Setting your HLASM processing options

When your HLASM copybooks fit into any of the following scenarios, you must set your HLASM processing options so that File Manager can process them correctly.

- You have used DBCS characters in your copybooks.
- You want the DC, DS and DXB instructions to be aligned on the correct boundaries only if the duplication factor is 0.

To set your HLASM processing options:

1. From the Primary Option Menu panel, select **0 Settings**, then **5 Compiler**, and **3 HLASM**.

   The HLASM Processing Options panel is displayed.
2. If you have used additional SYSLIBS, enter the names of the data sets in the **Additional SYSLIB data set** fields.

   Members included by COPY directives must be found in the SYSLIB concatenation. When you specify a copybook on a File Manager panel to create a new template (or update an existing template), the SYSLIB concatenation consists of the PDS of the copybook, plus up to ten additional PDSs that you can specify on this panel. These PDSs are searched in order (PDS of the copybook, followed by these additional PDSs, 1–10.)

   Note: CA-Panvalet libraries cannot be specified as additional SYSLIB sets. Further, if a CA-Panvalet Library is the main copybook library, then no additional SYSLIB data sets may be specified.
3. If your copybook contains DBCS characters, select the **DBCS** processing option.
4. If your HLASM copybook is compiled with NOALIGN, then select the **NOALIGN** processing option.
5. Additional compiler options which will be added via the *PROCESS statement.
6. To set the maximum acceptable return code for a compile, enter a value in the **Maximum Return Code to be accepted from compiler** field.

   When using the online panels to generate a template from a copybook, if the compiler returns a warning/error code greater than the specified number, File Manager displays a pop-up menu in which you can make decisions about further processing.

   When generating a template during the processing of a batch function, if the compiler returns a warning/error code greater than the specified number, File Manager halts the running of the function.
7. Press the Exit function key (F3) to save the changes you have made and exit the panel.

Related topics

"Using HLASM copybooks with File Manager" on page 138

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

Setting your PL/I processing options

When your PL/I copybooks fit into any of the following scenarios, you must set your PL/I processing options so that File Manager can process them correctly.

- You have used %INCLUDE directives to include other members that do not belong to the same PDSs as the copybooks.
- You have used DBCS characters in your copybooks.
- Your PL/I copybook contains declarations of 63-bit binary data.

- Your PL/I copybook declarations are not unaligned.
- Your PL/I copybook contains declarations of 31-digit decimal data.

To set your PL/I processing options:

1. From the Primary Option Menu panel, select **0 Settings**, then **5 Compiler**, and **4 PL/I**.

   The PL/I Processing Options panel is displayed.
2. If you have used additional SYSLIBS, enter the names of the data sets in the **Additional SYSLIB data set** fields.

   Members included by INCLUDE directives must be found in the SYSLIB concatenation. When you specify a copybook on a File Manager panel to create a new template (or update an existing template), the SYSLIB concatenation consists of the PDS of the copybook, plus up to ten additional PDSs that you can specify on this panel. These PDSs are searched in order (PDS of the copybook, followed by these additional PDSs, 1–10.)

   Note: CA-Panvalet libraries cannot be specified as additional SYSLIB sets. Further, if a CA-Panvalet Library is the main copybook library, then no additional SYSLIB data sets may be specified.
3. If your copybook contains DBCS characters, select the **GRAPHICS** processing option.
4. If your copybook contains signed 64-bit binary data fields, select the **63 bit binary** processing option.
5. If your copybook declarations are not aligned, select the UNALIGNED option.
6. If your copybook contains declarations of 31-digit decimal data, select the 31 digit decimal processing option.
7. Additional compiler options which will be added via the *PROCESS statement.
8. To set the maximum acceptable return code for a compile, enter a value in the **Maximum Return Code to be accepted from compiler** field.

   When using the online panels to generate a template from a copybook, if the compiler returns a warning/error code greater than the specified number, File Manager displays a pop-up menu in which you can make decisions about further processing.

   When generating a template during the processing of a batch function, if the compiler returns a warning/error code greater than the specified number, File Manager halts the running of the function.
9. Press the Exit function key (F3) to save the changes you have made and exit the panel.

Related topics

"Set PL/I Processing Options panel (option 0.5.4)" on page 656

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

Overriding Compiler Options

The current compiler options are stored in a template at the time the template is created. This allows subsequent updating of templates to be performed using the original compiler options.

To override the compiler options stored in a template:

1. Select the **Compiler Language Selection** option from the **Options** pull-down menu.

File Manager displays the Compiler Language Selection panel.

2. Select the **Override compiler options for template update** option.

   Using this option causes the template update to override the compiler options stored in the template with the currently specified compiler options. It also ensures the generation of the compiler parameters for the Batch JCL generated when the **Batch execution** option is selected.

Note: If the template does not contain compiler options, then the compiler options used are those currently specified for the execution or the relevant defaults.

Related topics

"Template Update Utility panel" on page 693

Determining the search order for your copybooks

The **Preserve copybook library** option allows you to determine the search order File Manager uses when looking for a specified copybook.

To determine the search order for your copybooks:

1. Select the **Compiler Language Selection** option from the **Options** pull-down menu.

   File Manager displays the Compiler Language Selection pull-down menu.

2. Select the **Preserve copybook library** option.

   Selecting this option causes the template update to attempt to locate the copybook in the data set in which it was originally found.

   Not selecting this option, or if the copybook cannot be located, causes the template update to search the library data sets in order for the first occurrence of the copybook.

The option only applies to template update.

You can also access the **Preserve copybook library** option on the Compiler Language Selection panel.

Related topics

"Template Update Utility panel" on page 693
"Compiler Language Selection panel" on page 470

Specifying or updating the source definition for a copybook template

The following sections describe how to specify or update the source definition for single and advanced copybook source definitions

Single copybook source definitions

- **Function entry panels**:

  Specify the data set name and member name in the **Copybook or Template** section for the copybook you require.

  Note: You can enter a generic member name to display a member list.

  Select the process option **1. Above** to use the copybook with the function. To edit the copybook template prior to the function, select the **Edit template** option and ensure you do *not* specify S for **Type**.

- **Workbench (Option 7.1)**:

Specify the data set name and member name in the **Copybook** section and issue the command you require *without* the **Advanced copybook selection** option selected.

Note: The CM and MT commands refer to a template only. The E command only refers to the copybook if the template data set and member name are blank.

Advanced copybook source definitions

- **Function entry panels**:

  Specify the **Copybook or Template** data set name and member name with either an existing template or copybook member and select the **Edit template** option with **Type** set to S. This takes you to the advanced copybook selection panels which allow you to change the source definitions and create a new template or update an existing template.

- **Workbench**:

  Select the option **Advanced copybook selection** and use the CC command with a copybook data set name and member name to create a new template, or the U prefix command with a template data set name and member name to update the source definition for an existing template.

  Note: You can also use the E command with the copybook data set name and member name, provided the template data set and member names are left blank.

Note: See "Advanced copybook selection" for a description of what you can do when specifying advanced copybook source definitions for templates.

If the **Edit template** option is used from a function entry panel with a copybook source definition, you are prompted to provide a template data set name (and, for partitioned data sets, a member name) to save the template that has just been created. Templates can be stored in sequential, PDS or PDSE data sets only.

Related topics

"Field Selection/Edit panel" on page 565
"Record Type Selection panel" on page 636
"Template Save pop-up panel" on page 697
"SAVE primary command" on page 812
"SAVEAS primary command (templates)" on page 813
"END primary command" on page 762

Advanced copybook selection

This section describes the method for specifying an advanced copybook source definition. To navigate to the panels mentioned, see "Advanced copybook source definitions."

1. The Library List panel is displayed as the initial panel when creating a new template. You can display this panel with the LIBLIST command from the Copybook Selection panel.

   You can specify up to 12 data set libraries. The data sets may be PDSs, PDSEs, CA-Panvalet libraries, or other library management system libraries. You can specify multiple PDSs, PDSEs, CA-Panvalet and other library management system libraries but they must be either all PDSs or PDSEs, or all CA-Panvalet, or all the same library management system libraries. You cannot mix library types.

2. Press the Exit function key (F3). The Copybook Selection panel is displayed initially for new templates where the member has not been supplied (after the

Library List panel), for new templates where a generic member name was specified on the Copybook section, or when the U prefix command was specified with a template data set name and member name to update the source definition for an existing template.

3. Type an "S" in the **Sel** field for each member that you want to select.

   You can browse (B), edit (E), or view (V) each of the members on this list before selecting the ones you want.

   Note:

   a. All members must contain copybook definitions in the same language.
   b. If you specify a member that consists of an entire program, then you should use the X prefix command to specify a statement range to extract the field definitions you are interested in, otherwise you may get compile errors when mixing with other members.
   c. You can only edit members in a PDS or PDSE data set.
   d. Library members may not be packed by ISPF.

4. When you have selected all of the members you want, press the Exit function key (F3). The selected members are listed in the Copybook Selection panel.

   From the Copybook Selection panel, you can display (and modify if necessary) the current library list by entering the LIBLIST primary command.

5. Edit the list of members to suit your requirements. You can copy or move members, insert or delete members and create new 01-level fields for any member. You can also specify redefines or range values for a member

   To add a new member:

   a. Enter the I prefix command to insert a new row.
   b. Type an asterisk or member name pattern in the Member field, then press Enter
   c. Select the new member name or names from the list. The list is not affected by any pattern that you have entered on the Template Workbench, it is a new list of all members in all selected databases, filtered by the pattern entered on the Copybook Selection panel.
   d. Press the Exit function key (F3). The selected members are inserted into your list on the Copybook Selection panel.

   To create a new 01-level field for a member:

   a. On the Copybook Selection panel, type a Y in the **01** field adjacent to the member.
   b. In the adjacent **Field Name** field, type a name for your 01-level element. This name must follow the naming conventions for the language of the copybook. If the name is omitted, File Manager generates a random name when the template is compiled.

   To extract a subset of the copybook to be compiled into the template:

   a. From the Copybook Selection panel, type an X in the **Cmd** field next to the required copybook member to display the Redefines / Range Specifications pop-up panel.
   b. In the **From statement** field, specify the starting line number for the extract.
   c. In the **To statement** field, specify the ending line number for the extract.
   d. In the **From string** field, specify the string that File Manager is to search the copybook for. The first statement that contains the string begins the extract.
   e. In the **To string** field, specify the string that File Manager is to search the copybook for. The first statement that contains the string ends the extract.

See the panel definition for a complete list of the prefix commands available.

6. To generate multiple record layouts for a source that contains more than one record definition within a single level-01 structure:
   a. From the Copybook Selection panel, type an X in the **Cmd** field next to the required copybook member to display the Redefines / Range Specifications pop-up panel.
   b. In the **Level** field, specify the source level value to identify the COBOL REDEFINES or the PL/I UNION clauses for generating new record layouts. File Manager creates a separate record layout for each REDEFINES or UNION clause at the specified level. After the first matching REDEFINES or UNION clause is found, File Manager creates new layouts for the latter and subsequent REDEFINES or UNION clauses for the same level and start location. Header and trailing data items are included in each record layout.

      Note: This field is ignored for HLASM compiles.
   c. In the **Field name** field, specify the target field name of the COBOL REDEFINES, PL/I UNION, or HLASM ORG clause, that identifies the COBOL REDEFINES, PL/I UNION, and HLASM ORG statements that are to be used to direct File Manager to create new layouts.

      This is an alternative way of identifying COBOL REDEFINES, PL/I UNION, and HLASM ORG statements that require new layouts to be generated. You can specify both **Level** and **Field name** values; File Manager checks both when generating new layouts.
   d. To adjust the offset value for each layout so that the starting location is the REDEFINES, UNION, or ORGfield start location, enter a "/" in the **Set offset** field.

      Use this option when your record layouts do not include the header data items in the structure. You can only select this option in conjunction with the previous fields.
   e. **COBOL level change**: This is a COBOL ONLY option which changes the way the **Level** value is processed. If you set this option, every occurrence of the specified **Level** value is changed to '01" before the compile step, irrespective of whether the data item has a REDEFINES clause.

      Use this option with care as it may result in incorrect offsets to data items, or compile errors, due to the structure being changed from its intended programmable form.
7. To compile the specified copybooks from the Copybook Selection panel, and either update an existing template or create a new template, use the UPDATE primary command or the U prefix command.
8. To edit an existing template built from the copybooks specified on the Copybook Selection panel, use the EDIT primary command or the E prefix command. If you have made any changes, File Manager compiles the copybooks and updates the template.
9. When your Copybook Selection list is completed, press the Exit function key (F3).

   The compiler attempts to create the template for you.

   If the compilation is unsuccessful, the Template Not Built pop-up panel is displayed, offering you the following choices:

   1. View the compilation listing.
   If you select this option, the compilation listing is displayed so that you can determine the cause of the error. When you have finished viewing

the listing, press the Exit function key (F3) to return to the Copybook Selection panel, correct the problem, then try to exit the panel again.

2. Abort processing.
Returns you to the Copybook Selection panel, where you can correct the problem, then try to exit the panel again.

3. Retry (recompile copybook)
Attempts to submit the same details again. If the compilation had failed because a member was temporarily unavailable, this may resolve the problem.

4. Retry (with alternate compiler)
Attempts to submit the same details again but with the alternative compiler. If the compilation had failed because the File Manager default compiler was set to the wrong compiler for this copybook language, this may resolve the problem.

Note: This choice does not apply to HLASM.

If the compilation is successful, you are prompted to save it to the template member specified on the Template Workbench panel. If you had supplied a member name pattern or left the **Template: Member** field blank on that panel, the Template Member Selection panel is displayed. You can select a name from this list and press Enter to overwrite the existing selected member, or you can press the Exit function key (F3) and you are prompted to name a new member.

Related topics
"Template types and structure" on page 127
"Template Workbench panel" on page 698
"Library List panel" on page 588
"Copybook Selection panel" on page 505
"LIBLIST primary command" on page 785
"UPDATE primary command" on page 827
"EDIT primary command" on page 761
"Redefines / Range Specifications panel" on page 638

Creating a copybook template with the Template Workbench

To create a copybook template with the Template Workbench:

1. Open the Template Workbench panel by performing either of these actions
   -
     a. Selecting option **7. Templates** from the Primary Option Menu panel to display the Copybook and Template Utility functions panel, and then
     b. Selecting option **1. Workbench** from the Copybook and Template Utility functions panel to display the Template Workbench panel.
   - Issuing the TVIEW primary command on either the View or the Edit panel. If it is accessed from an editor session, the Template Workbench contains two additional options (**RC - Run using copybook** and **RT Run using template**).
2. In the **Data set name** field in the **Copybook** section, type the name of the data set in which your copybook is stored.

   Copybooks can be stored in PDS, PDSE or CA-Panvalet libraries. Library members may not be packed by ISPF.
3. If you know it, type the Member name of the copybook into the relevant field or specify blanks or a pattern for the member name to have a member selection list displayed. You can then select the required member from the list.

4. In the **Data set name** field in the **Template** section, type the name of an existing data set, into which you want the new template to be saved.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.
5. In the **Member** field of the **Templates** section, type the name for your new template.

   If you want to copy the copybook to an existing template (which overwrites the existing template), specify an existing member name or specify blanks or a pattern to display the Member Selection panel. You can then select the required member from the list.
6. On the **Command Line**, type the `CC` command and press Enter.

   File Manager validates the copybook by compiling it with your default compiler and, if successful, generates and saves the template. A message is displayed in the upper right corner of the panel, to indicate the success or failure of the process.

Note: If you do not provide a template data set name, File Manager creates a temporary template, which you can then edit. Before you close the Template Workbench, you can save the template by specifying a template name and entering the SAVE command. Otherwise the template is discarded.

Related topics
"Template Workbench panel" on page 698

Setting up a template to be used with segmented data

Templates that contain more than one record type can be set to be used with segmented data by entering a "/" in the **Template for segmented data** field.

When you are viewing or editing data using a segmented data template, the following restrictions apply to template editing:

- You cannot update the **Template for segmented data** field in the Record Type Selection panel (accessed with the TEDIT command).
- You cannot change the Record Identification Criteria, Related Identification Criteria, or Selection Criteria set in the Field Selection/Edit panel (accessed with the TEDIT command and then the E command on the Record Type).
- You cannot use the TVIEW command (which would otherwise display the Template Workbench, allowing you to select or create a different template).
- The length of a segment is calculated as the length of the matching structure, unless you provide a length field. Any binary or alphanumeric field less than 4 bytes in length can be set as the length field by editing the field attributes. If a length field is provided, then File Manager uses the binary value provided as the segment length and adjusts the structure length accordingly.

Creating a template from a model template

You can use the Template Workbench to create copies of an existing copybook or dynamic template. The original template serves as a model, providing template information such as record selection criteria that you can then edit in the new copy.

1. You can choose to specify record type definitions that do not contain any Record Identification Criteria. If File Manager cannot match a segment (using Record Identification Criteria) with any of the other record types, it uses the first record type without any Record Identification Criteria, whose length is less than or equal to the remaining length on the record, as a default segment.

To create a template from an existing (model) template:

1. Open the Template Workbench panel by performing either of these actions
   -
     a. Selecting option **7. Templates** from the Primary Option Menu panel to display the Copybook and Template Utility functions panel, and then
     b. Selecting option **1. Workbench** from the Copybook and Template Utility functions panel to display the Template Workbench panel.
   - Issuing the TVIEW primary command on either the View or the Edit panel. If it is accessed from an editor session, the Template Workbench contains two additional options (**RC - Run using copybook** and **RT Run using template**).
2. In the **Data set name** field in the **Model Template** section, type the name of the data set in which your existing template is stored.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.
3. If you know it, type the Member name of the template into the relevant field or specify blanks or a pattern for the member name to have a member selection list displayed. You can then select the required member from the list.
4. In the **Data set name** field in the **Template** section, type the name of an existing data set, into which you want the new template to be saved.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.
5. In the **Member** field of the **Templates** section, type the name for your new template.

   If you want to copy the template model to another existing template (which overwrites the existing template), specify an existing member name or specify blanks or a pattern to have a member selection list displayed. You can then select the required member from the list.
6. On the **Command Line**, type the `CM` command and press Enter.

   File Manager generates and saves the template. A message is displayed in the upper right corner of the panel, to indicate the success or failure of the process.

Note: If you do not provide a template data set name, File Manager creates a temporary template, which you can then edit. Before you close the Template Workbench, you can save the template by specifying a template name and entering the SAVE command. Otherwise the template is discarded.

Related topics

"Template Workbench panel" on page 698

Creating dynamic templates

You create dynamic templates by building field definitions for your data set or by converting an existing copybook template to a new dynamic template.

Creating dynamic templates on the fly

When you create a dynamic template, for each field definition you build, you specify the field definition details and, optionally, selection criteria that File Manager uses to filter records. When you specify selection criteria for a field, you do not necessarily have to supply all the field definition details as File Manager supplies default values.

When you are building the definitions *without* selection criteria, you must supply the starting column, length and type of the data set fields that you wish to operate upon.

When you are building the definitions *with* selection criteria, you must supply the starting column and value for the data set fields that you wish to operate upon. If you do not supply the length of a field, File Manager calculates it from the value you specify. Similarly, if you do not supply the data type or operator for a field, File Manager uses default values.

In either case, you can, optionally, supply your own field name to identify the information.

You can edit these field definitions at any time.

Note: If you define a field with a type of VC (varying character), VG (varying graphic), or VD (varying DBCS), without also defining a 2-byte binary field immediately preceding it, when generating the template, File Manager reduces the length of the field you have defined by 2 bytes and inserts a 2-byte binary length field with the field name, LEN, immediately before it.

1. In the **Copybook/template usage** field of any relevant panel, select option **4. Create dynamic** and press Enter.

   The Dynamic Template panel is displayed.
2. For each field in the data set that you want to display *without* specifying selection criteria:
   - Specify values in the **Start**, **Length**, and **Type** fields.
3. For each field in the data set that you want to display and for which you want to specify selection criteria:
   - Specify a value in the **Start** field.
   - Specify a value in the **Value** field.
   - Optionally, specify a value in the **Length** field. If you leave this field blank, File Manager calculates the length from the value you specify in the **Value** field.
   - Optionally, specify a value in the **Type** field. (See note above for types VC, VG, and VD.) If you leave this field blank, it defaults to "AN" (alphanumeric).
   - Optionally, specify a value in the **Op** (Operator) field. If you leave this field blank, it defaults to "=".
4. If you wish, specify a Field Name for your field definitions.

   If this field is left blank, a name is generated in the form **@@DT***nn* where *nn* is the field number. Existing field names taking the form **@@DT***nn* are regenerated to ensure that *nn* always reflects the current field number.

   Note: If you are running a screen width of 115 or greater, the Field Name field can be displayed at the same time as the Start, Length and Type fields. However, at any width less than this, you need to toggle the display between Field Name and Start, Length and Type. The default is to display Start, Length and Type. To toggle the display, ensure that your cursor is not in a Value field and press the Right function key (F10) or Left function key (F11).
5. Press the Exit function key (F3). The Template Save pop-up panel is displayed.
6. Type the name of an existing data set in the **Data set name** field.

Templates can be stored in PDS or PDSE data sets, but cannot be stored in CA-Panvalet libraries.

7. Type a name for your new template in the **Member** field.
8. Press the Exit function key (F3). The template is saved and the next relevant panel (depending upon your entry point) is displayed.

Related topics

"Manipulating the display of fields in records" on page 162
"Dynamic Template panel" on page 529
"Template Save pop-up panel" on page 697

Creating dynamic templates from copybook templates

Instead of building field definitions manually, you can create a new dynamic template using the definitions in a copybook template. Once created, you can edit all aspects of the new dynamic template, including the field definitions.

1. Specify your copybook template on the relevant Entry panel and select the Edit Template option.
2. On the Field Selection/Edit panel, enter `2` to display the **Specify Record Selection by field** panel.
3. Type `MD` on the Command line and press Enter.

   The Dynamic Template panel is displayed, containing the list of field definitions generated from your copybook template. You can modify any of these values.

   Note: As all field definitions on a dynamic template are treated as Level 2 elementary items, you may need to delete any previous group level fields.
4. Choose one of the following actions:
   - Enter the `SAVE` command and then exit or cancel from the panel.

     If you save the dynamic template, you are prompted for a data set and a member name in which to store the new dynamic template.
   - Press the Cancel function key (F12) to return to using your copybook template, without saving the dynamic template.
   - Press the RunTemp function key (F6) to display your data with the temporary dynamic template (only valid if you have an active editor session).

     Tip: The `RunTemp function key (F6)` should not be used with the Template Workbench nor any function which has selected batch execution. Currently, this is not enforced and can lead to unpredictable results.
   - Exit from the panel and, when prompted by the Template Save panel, choose to save the template, run with a temporary template or cancel your changes.

You can also create a dynamic template from a copybook template while in an editor session. To do this:

1. Enter `CEDIT` on the Command line of the View or Edit panel. The Record Selection Criteria panel is displayed.
2. Type `MD` on the Command line and press Enter.

   The Dynamic Template panel is displayed, containing the list of field definitions generated from your copybook template. You can modify any of these values.

3. Exit from the panel (press F3) and, when prompted by the Template Save panel, choose to save the template, run with a temporary template or cancel your changes.

When working from an active editor session, the following applies:

- When you return to the editor panel, the active copybook template is replaced with the temporary or saved dynamic template.
- If you return to the editor panel using a temporary dynamic template and then exit from the panel, you are not prompted to save the template and it is discarded.
- You can re-edit the dynamic template by entering the `DEDIT` command. This would give you an opportunity to save a temporary template.

Related topics
"CEDIT primary command" on page 747
"DEDIT primary command" on page 758

Creating corresponding templates from selected copybooks

You can create multiple templates from a PDS of copybooks by using the Template Build Utility. The utility builds one template from each selected copybook. The template name is derived from the input member name and any output member mask that you have specified. If you do not supply a template member mask, then the template name is the same as the copybook name.

You can run the Template Build Utility in batch or foreground.

To create multiple templates from a PDS of copybooks:

1. Navigate to the Template Build Utility panel by:
   a. Selecting option **7. Templates** from the Primary Option Menu panel to display the Copybook and Template Utility functions panel, and then
   b. Selecting option **3. Build** from the Copybook and Template Utility functions panel to display the Template Build Utility panel.
2. In the Copybook **Data set name** field, specify the data set name of the copybook PDS(E). The name you specify must be a PDS(E), except for vendor-managed library systems such as Panvalet, CA/Librarian, and so on.
3. In the Copybook **Member** field, specify the name of a partitioned data set member that contains a copybook.

   If you leave the **Member** field blank, or specify a member name pattern, File Manager displays a member name list. You can then select the required member by entering S in the **Sel** field for the appropriate member.

   If you select **Advanced member selection**, then this name constitutes the first name on the member range panel.
4. In the Template **Data set name** field, specify the PDS or PDSE that identifies where the templates are to be stored.
5. In the Template **Member mask** field, optionally specify a pattern to rename members in the output partitioned data set based on the member names in the input partitioned data set.

   Note: The **Member mask** works in the same way as the mask you can use when copying data sets. For an explanation of how you can specify a member name mask in the form of a member name pattern, see "Renaming members as you copy" on page 258.

6. To run the Template Build Utility in batch, select the **Batch execution** processing option.

   Batch processing produces up to two reports: the Data Set List report (only produced if a concatenated data set is used for input), and the Template Build Report. Examples of each report are shown below.

```
Data set name                                  Lib
-----------------------------------------------------
FMNUSER.COBOL2                                  1
FMNUSER.COBOL                                   2
```

Figure 54. Data set list report

```
          Template Build Report
Copybook   Lib Template    Status
-------------------------------------------------------------------------------
DITTST1      1 DITTST1     Replaced
DITTST3      1 DITTST3     Replaced
DITTST7      1 DITTST7     Compile error
FMNTST1      2 FMNTST1     Created

4 members read : Template : 1 Created  2 Replaced  1 Errors
```

Figure 55. Template Build report

   Note: The **Lib** column is only displayed when more than one input data set is found.

7. To replace templates that exist with the same name, select the **Replace existing templates** processing option.
8. To specify a range of members to be selected, rather than a specific or generic member name, select the **Advanced member selection** processing option.
9. To run without showing the member selection list, select the **Skip member list** processing option. This option is ignored if errors are found (such as duplicate member names due to the rename mask).

Related topics

"Template and Copybook Utility functions panel" on page 687
"Template Build Utility panel" on page 688
"Member Selection panel" on page 600

Editing a template

You can perform template editing:

- From any panel that has the **Edit template** option
- From the Template workbench
- By using the appropriate template primary commands from an editor session.

  Template commands from within an editor session may be restricted in function, depending on the type of Edit or View being performed.
- By using the E command from the template update utility member list (option 7.4).

To perform a template edit, you can either select the **Create dynamic** option or, on panels with the **Edit template** option, you can select that option and set **Type** to one of the following:

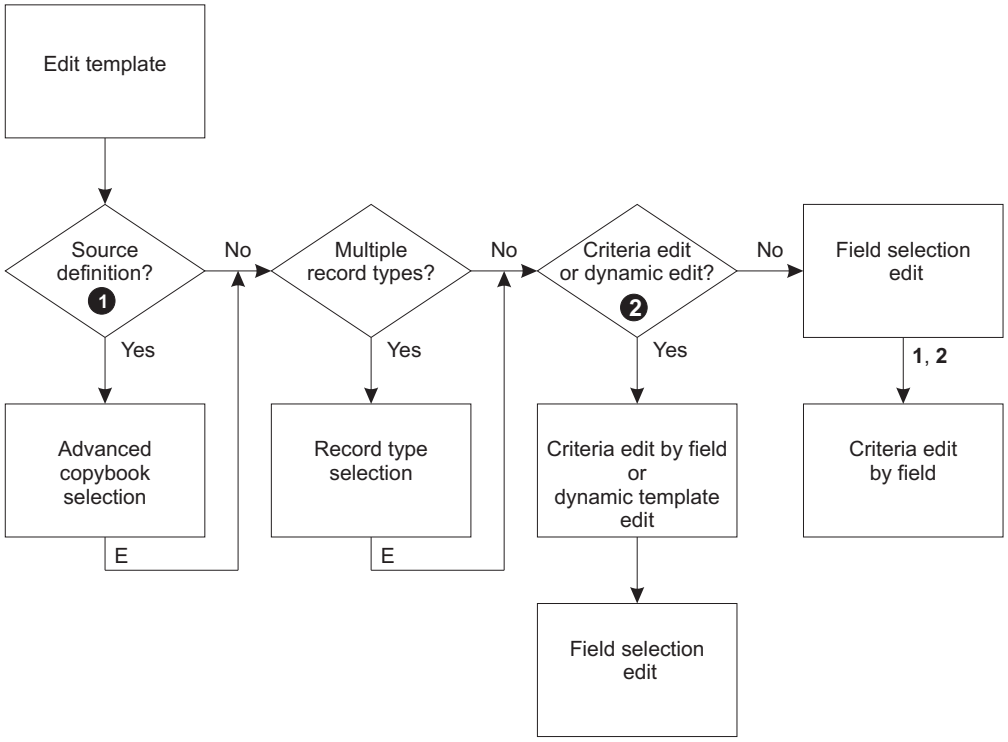
Blank Normal record type or field/selection edit for single layout copybooks or templates.

S To edit the source definition using the advanced copybook selection process.

1 To perform criteria edit by field for identification criteria before the field/selection edit. This provides a fast path when you just want to specify criteria by field.

2 To perform criteria edit by field for selection criteria before the field/selection edit. This provides a fast path when you just want to specify criteria by field.

The options above are ignored for dynamic templates. If you specify **Create dynamic** or edit an existing dynamic template, then File Manager always takes you to the dynamic edit panel.



1 **Source definition:**
- **Type** "S" specified and **Edit template** option selected, or
- CC or U command entered on Template Workbench panel with:
  - **Advanced copybook selection** option selected, or
- E command entered on Template Workbench panel with:
  - Copybook data set name specified, and
  - No template data set name specified, and
  - **Advanced copybook selection** option selected

2 **Criteria edit or dynamic edit:**
- **Type** "1" or "2" specified and **Edit template** option selected, or
- Template is a dynamic template, or
- CE or DE command entered during an editor session

Figure 56. Editing a template: task flow

To edit a template from a supported panel:

1. In the **Copybook or Template** section, specify the data set name of a copybook or template.

Templates can be stored in sequential (PS), PDS or PDSE data sets. CA-Panvalet libraries can only be used as input data sets for copybooks.

2. In the same section, specify the Member name or leave the Member name field blank to choose from the list of data set members.
3. Set the **Copybook/template** flag to **Above** or **Create dynamic**.
4. If you have not selected **Create dynamic**, then select the **Edit template** option and set **Type** to determine the point at which you want to perform template editing:

   Blank Normal record type or field/selection edit for single layout copybooks or templates.

   S To edit the source definition using the advanced copybook selection process.

   1 To perform criteria edit by field for identification criteria before the field/selection edit. This provides a fast path when you just want to specify criteria by field.

   2 To perform criteria edit by field for selection criteria before the field/selection edit. This provides a fast path when you just want to specify criteria by field.
5. Make the required editing changes and do *one* of the following:
   - When you are creating a new template, pressing the Exit function key (F3) displays the Template Save pop-up panel.
   - Type SAVE on the Command line and press Enter. You can then choose to make further changes, continue to the next relevant panel or return to your entry panel.
   - Type SAVEAS on the Command line and press Enter. Specify the data set (and member name if a PDS) of the new template and press the Exit function key (F3). You can then choose to make further changes, continue to the next relevant panel or return to your entry panel. The new template is used in place of the original for the current function.
   - Press RunTemp function key (F6) to display the next relevant panel using the changes you have made but without saving those changes.
   - Press Cancel function key (F12) to return to your entry panel without saving the changes.

To edit an existing copybook template from an editor panel:

1. Type one of the following commands on the Command line and press Enter.

   TEdit Template Edit: If your copybook has only one record type, the Field Selection/Edit panel is displayed. From here, you can enter 1 or 2 to access the Record Identification Criteria or Record Selection Criteria panels.

   If your copybook has two or more record types, the Record Type Selection panel is displayed. In the **Command** field adjacent to the record type that you would first like to edit, type an E and press Enter. The Field Selection/Edit panel for that record type is displayed.

   CEdit (or CEdit SEL or DEdit)
   Criteria Edit (Selection): The CE, CE SEL and DE commands are all synonymous in this situation and display the Record Selection Criteria panel, showing the fields in the currently displayed record type. From here, you can enter 1 to access the Field Selection/Edit panel. If your copybook has two or more record types, the Record Type Selection

panel displays at this point. In the **Command** field adjacent to the record type that you would first like to edit, type an `E` and press Enter. The Field Selection/Edit panel is displayed.

CE ID Criteria Edit (Identification): The Record Identification Criteria panel is displayed. From here, you can enter 1 to access the Field Selection/Edit panel. If your copybook has two or more record types, the Record Type Selection panel is displayed at this point. In the **Command** field adjacent to the record type that you would like to edit, type an `E` and press Enter. The Field Selection/Edit panel is displayed.

2. Make the required editing changes and do *one* of the following:
   - Press the Exit function key (F3) to save the changes and display the next relevant panel (for example, the View panel). The message "Template saved" appears in the upper right corner of the panel.

     Note: Unlike when creating a new template, the Template Save pop-up panel is not displayed at this point. The previous version of the template is automatically replaced by the changes you have made.
   - Type SAVE on the Command line and press Enter. You can then choose to make further changes, continue to the next relevant panel or return to your entry panel.
   - Type SAVEAS on the Command line and press Enter. Specify the data set (and member name if a PDS) of the new template and press the Exit function key (F3). You can then choose to make further changes, continue to the next relevant panel or return to your entry panel. The new template is used in place of the original for the current function.
   - Press RunTemp function key (F6) to display the next relevant panel using the changes you have made but without saving those changes.
   - Press Cancel function key (F12) to return to your entry panel without saving the changes.

To edit an existing dynamic template from an editor panel:

1. Type one of the following commands on the Command line and press Enter.

   TE Template Edit: The Field Selection/Edit panel is displayed. However, you cannot change the Field Name, Start, Length or Type fields from this panel, nor can you change the record selection criteria. To perform these tasks, enter 1 to display the Dynamic Template panel.

   DE (or CE, CE SEL or CE ID)
   Dynamic Edit: The CE, CE SEL and DE commands are all synonymous in this situation and display the Dynamic Template panel.
2. Make the required editing changes and do *one* of the following:
   - Press the Exit function key (F3) to save the changes and display the next relevant panel (for example, the View panel). The message "Template saved" appears in the upper right corner of the panel.

     Note: Unlike when creating a new template, the Template Save pop-up panel is not displayed at this point. The previous version of the template is automatically replaced by the changes you have made.
   - Type SAVE on the Command line and press Enter. You can then choose to make further changes, continue to the next relevant panel or return to your entry panel.
   - Type SAVEAS on the Command line and press Enter. Specify the data set (and member name if a PDS) of the new template and press the Exit function

key (F3). You can then choose to make further changes, continue to the next relevant panel or return to your entry panel. The new template is used in place of the original for the current function.

- Press RunTemp function key (F6) to display the next relevant panel using the changes you have made but without saving those changes.
- Press Cancel function key (F12) to return to your entry panel without saving the changes.

Related topics

"Manipulating the display of fields in records" on page 162
"Dynamic Template panel" on page 529
"Record Type Selection panel" on page 636
"Field Selection/Edit panel" on page 565
"SAVE primary command" on page 812
"SAVEAS primary command (templates)" on page 813

Updating one or more templates

The Template Update Utility allows you to update one or more templates in either foreground or batch.

You can filter template member selection to select only templates that reference a certain copybook or copybooks. You can specify a generic copybook filtering name.

1. Select option **7. Templates** from the Primary Option Menu panel to display the Copybook and Template Utility Functions panel.
2. Select option **4. Update** from the Copybook and Template Utility Functions panel to display the Template Update Utility panel.
3. On the Template Update Utility panel, enter the details to select the templates you want to update:
   - In the Template **Data set name** field, specify a fully-qualified or generic data set name to identify the data set (which must be partitioned).
   - In the Template **Member** field, you can specify the name of a member of a partitioned data set.
   - In the Template **Copybook filter** field, you can specify up to 4 member names or patterns to be used as a filter so that only templates referencing those copybooks, or copybooks that match the member patterns, are selected for processing.
4. In the Output Template **Data set name** field, you can specify a fully-qualified or generic data set name to identify the output data set where the updated templates are to be stored. If you leave this field blank, then the update takes place on the input data set.
5. In the Output Template **Member mask** field, you can specify a rename mask so that the updated templates are stored under an alternative name.
6. Select the processing options:
   - To run the function in batch, select **Batch execution**.
   - To replace like-named members in the output partitioned data set, select **Replace members**.
   - To specify a range of members to be selected, rather than a specific or generic member name, select **Advanced member selection**.
   - To use the library data sets specified on the entry panel select **Use library data sets**.

- To run without showing the member selection list, select **Skip member list**. This option bypasses the member list panel and processes all the qualifying members. If an error (like a duplicate output name) results from the specified parameters, then the member list panel is displayed with the errors highlighted.
- To run the function without saving the resulting members, select **Check mode - no update**.
- If necessary, select the **Preserve copybook library** option to ensure that, if a copybook still exists in the library that it was previously found in and that library is in the list that the update is using, then that version of the copybook is used if:
  - This option is not selected, or
  - The copybook no longer exists in the library it was previously found in, or
  - That library is not in the list the update is using then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

7. If you have selected the **Use library data sets** option, you can specify up to twelve copybook library names in the Library data sets **Data set name** fields.

   The copybook library names referenced in the template are changed to the copybook library names specified in these fields. The new library data sets are used to locate and compile the copybooks during the update process, so you need to ensure that all the copybooks referenced in all the templates selected are available in the libraries you specify to avoid update errors occurring.

8. Press Enter. Unless you have selected the **Skip member list** processing option, File Manager displays a member list of templates like those shown in Figure 57. Press F11 (Right) to scroll right to display the Description column, and F10

```
 Process   Options   Help
 ┌──────────────────────── Library List ──────────────────────────┐ ──────
File Manager              Template Update Utility               Row 1 of 104

Input data set  'FMN.RFM0047.TEMPLATE'
Update data set 'FMN.RFM0047.TEMPLATE'

Sel Name     Prompt       Created     Updated              Lang   Ver
    ABEND                 2004/02/11  2004/02/11 09:09:15  COBOL    1
    BIGCHAR               2004/07/14  2006/06/28 16:12:31  COBOL    1
    BIGCOPY               2003/12/17  2003/12/17 14:39:16  COBOL    1
    BIGCOP2               2004/02/11  2004/02/11 10:12:10  COBOL    1
    BIGKSDS               2001/08/10  2001/08/16 13:54:23  COBOL    1
```

Figure 57. Member list of templates

   (Left) to return to the original display.

9. In the prefix area (**Sel**), type any of the following prefix commands and press Enter to perform the associated action:
   - To browse a template, type the B prefix command. The member is displayed using the ISPF Browse panel (not the File Manager Browse panel).
   - To invoke template editing, type the E prefix command. File Manager invokes template editing.
   - To invoke the template source definition edit and interactive update process for the template, type the U prefix command. File Manager displays the Copybook Selection panel.

- To select a template for updating, type the S prefix command. File Manager saves the updated template.

Related topics
"Template and Copybook Utility functions panel" on page 687
"Template Update Utility panel" on page 693

Manipulating the display of fields in records

File Manager offers a number of ways to manipulate how fields are displayed within records. Using the Field Selection/Edit panel, you can:

- Select fields
- Change the order in which fields are displayed
- Change field headings
- Access the Field Attributes panel to change individual field attributes

Selecting and deselecting fields

You can change the display of a record type to show selected fields only, for display and printing purposes. If you select fields, then only those fields are displayed or printed. If you do not select fields, then File Manager displays or prints all fields.

To select or deselect individual fields:

1. Edit your copybook or dynamic template in the Field Selection/Edit panel.
2. In the **Cmd** field adjacent to the field to be selected, type S and press Enter.

   The **SHE** column displays an "S" and the line becomes highlighted. For example, in Figure 58, the REC-TYPE and NAME fields are selected.

```
⋮
 ──────────────────────────────────────────────────────────────────────────────
 File Manager                Field Selection/Edit

 1 Specify Record Identification by field.
 2 Specify Record Selection      by field.

 Cmd Seq SHE Ref Rdf Field Name                                Type  Start Length
                    ****  Top of data  ****
 ___ ___         1     1 REC-TYPE01                             AN      1     80
 ___ ___ S       2      2 REC-TYPE                              AN      1      2
 ___ ___ S       3      2 NAME                                  AN      3     20
 ___ ___         4      2 EMPLOYEE-NO                           BI     23      2
 ---------------------- Record Identification Criteria -------------------------

⋮
```

Figure 58. Field Selection/Edit showing selected fields

3. To remove the selection status, type the S command in the **Cmd** field adjacent to a selected field. If you remove the selection status from all fields in the record type, File Manager reverts to its default behavior, which is to display or print all fields.

You can toggle between selecting and deselecting more than one line at a time by entering one of the following selection commands:

SS Selects and deselects a block of lines.

Enter SS twice: first in the **Cmd** field at the start of the block you want to select or deselect, and second, at the end of the block. File Manager toggles the selection status of each line in the selected block.

S*n* Selects and deselects *n* lines.

Enter S*n* in the **Cmd** field of the first line you want to select or deselect. File Manager toggles the selection status of each line, starting from the first line selected for *n* lines.

S* Selects and deselects a block of lines, starting from the first line selected and then all subsequent lines.

Enter S* in the **Cmd** field of the first line you want to select or deselect. File Manager toggles the selection status of each line, starting from the first line selected to the last line.

Related topics
"Field Selection/Edit panel" on page 565
"Editing a template" on page 156

Changing the field display order

By default, fields are displayed in the order in which they occur in the data set. However, by entering numbers in the **Seq** field (on the Field Selection/Edit panel), you can override this order.

Any fields with sequence numbers are displayed before fields without sequence numbers. Fields with sequence numbers are displayed in the relative order of the sequence numbers. Sequence numbers do not need to start at 1, or be consecutive. The remaining fields (without sequence numbers) are displayed in their default order.

Holding columns when scrolling left or right

You can hold or "free" fields so that, when you scroll left or right in TABL display format, those fields are always displayed on the left of the screen regardless of how far you scroll. To hold a field, enter `H` in the **Cmd** field.

When you hold a field, "H" appears in the middle position of the **SHE** column on the Field Selection/Edit panel, and "1" appears in the **Seq** column to show it is the first held field. If you hold a second field, "2" appears in the **Seq** column to show it is the second held field, and so on.

Note: The sequence numbers of held fields are independent of any other sequence number you might add. Held fields always appear as the leftmost columns on the screen (in the order that you held them when editing the template) and take priority (as far as the left-to-right display is concerned) over other fields.

Toggling the hold status of more than one field at a time

You can toggle between holding or "freeing" more than one field at a time by entering one of the following hold commands in the **Cmd** field of the Field Selection/Edit panel:

HH Holds or "frees" a block of fields.

Enter `HH` twice: first in the **Cmd** column at the start of the block of fields you want to hold or "free", and again at the end of the block. File Manager toggles the hold status of each column in the selected block.

H*n* Holds or "frees" *n* fields.

Enter H*n* in the **Cmd** field of the first field you want to hold or "free". File Manager toggles the hold status of each field, starting from the first selected field and continuing for *n* fields.

Changing field headings

By default, File Manager uses the field names (from your original copybook or from your dynamic template) as field headings in SNGL and TABL display format. You can override this by specifying your own field headings.

To change the field heading:

1. Enter the `E` (edit) line command in the Cmd field adjacent to the field that you want to rename.
2. Enter the new field heading in the **Heading** field on the Field Attributes panel.
3. Press the Exit function key (F3) to save your change and return to the Field Selection/Edit panel.
4. Repeat for all field headings you want to change.

The field heading may determine the width of the display column. The default output width is the maximum of the number of characters needed to show the field heading (or field name, if no heading is specified), and the number of characters needed to show the value of the field. The following figures demonstrate how changing a field heading can change the display width of the field.

```
:
REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY   MONTH(1)
#2       #3                          #4      #5       #6         #7
AN 1:2   AN 3:20                BI 23:2 BI 25:2  PD 27:4    BI 31:4
<>       <---+----1----+---->    <---+-> <---+-> <---+--> <---+---->
****  Top of data  ****
01       Grant Smith                7712      35    75000          6
01       Andrew Apple               6645      53    78500         30
:
```

Figure 59. Data set displayed using default field names

```
:
Type   NAME                   Emp #     AGE   SALARY     Mth(1)     Mth(2)
#2     #3                        #4      #5       #6         #7         #7
AN 1:2 AN 3:20              BI 23:2 BI 25:2  PD 27:4    BI 31:4    BI 35:4
<>     <---+----1----+----> <---+-> <---+-> <---+--> <---+----> <---+---->
****  Top of data  ****
01     Grant Smith             7712      35    75000          6         15
01     Andrew Apple            6645      53    78500         30         22
:
```

Figure 60. Data set displayed using field headings

Note: While the width of the Employee-No field has reduced, the width of the Month column has not changed. This is because the Month field has a Type of BI and Length of 4. The maximum value that can be entered into a field of this type has 9 digits, therefore, the default output width is 9 characters. To reduce the width of the Month fields, you would need to modify the field attributes.

Changing field attributes

Field formatting attributes, such as the width of a field for TABL formatting or leading zero suppression for numeric fields, can be changed on the Field Attributes panel. You can also specify data create attributes on this panel . The panel comes in two flavors: one for alphanumeric fields and the other for numeric fields.

Selecting a field for attribute change

You select a field from the Field Selection/Edit panel (Figure 174 on page 566). To select a record for attribute change, enter `E` in the **Cmd** field adjacent to the required field. You can enter E against as many fields as you like. For each field you select, File Manager displays a Field Attributes panel.

The attributes you can change are:

- Output width
- Whether to display or suppress leading zeros on numeric fields
- Values to use for creating new records

The attributes are used to determine how to display or print a field.

Changing the attributes for an alphanumeric field

You use the alphanumeric Field Attributes panel to change attributes for an alphanumeric field or set the pattern for created fields.

To change the output width, enter the new width into the **Output width** field.

The output width is the number of character positions used by edit, view, and print panels to show the field in TABL display or print format. The default output width is the maximum of the number of characters needed to show the field heading (or field name, if no heading is specified), and the number of characters needed to show the value of the field, as determined from the field definition.

The minimum width allowed is six characters. The maximum is 30 or the field width + 10 (whichever is the greater).

You can set any field less than 4 bytes long to be the length field during a segmented template edit. You can specify the length as inclusive or exclusive. If a non-blank value is provided, then the segment length is determined from the binary value contained in the field, and for exclusive fields the field length is added to this value.

Changing the attributes for a numeric field

You use the numeric Field Attributes panel to change attributes for a numeric field or set the pattern for created fields.

To change the output width, enter the new width into the **Output width** field. The output width is the number of character positions used by edit, view, and print panels to show the field in TABL display or print format. The default output width is the maximum of the number of characters needed to show the field heading (or field name, if no heading is specified), and the number of characters needed to show the value of the field, as determined from the field definition, including a sign character and decimal point.

The allowable widths are six to 30 characters.

To display leading zeros on numeric fields in SNGL and TABL display format, enter YES in the **Leading zeros** field. The default is NO (suppress leading zeros).

The following figure demonstrates how field headings and output widths affect the column width display.

```
⋮
Type   NAME                   Emp #     AGE   SALARY Mth(1) Mth(2) Mth(3)
#2     #3                        #4      #5       #6     #7     #7     #7
AN 1:2 AN 3:20              BI 23:2 BI 25:2  PD 27:4 BI 31: BI 35: BI 39:
<>     <---+----1----+----> <---+-> <---+-> <---+--> <---+> <---+> <---+>
****  Top of data  ****
01     Grant Smith             7712      35    75000      6     15     42
01     Andrew Apple            6645      53    78500     30     22     46
⋮
```

Figure 61. Data set displayed using field headings and output width adjustment

You can set any field less than 4 bytes long to be the length field during a segmented template edit. You can specify the length as inclusive or exclusive. If a non-blank value is provided, then the segment length is determined from the binary value contained in the field, and for exclusive fields the field length is added to this value.

Displaying alphanumeric fields in long hexadecimal

You can change the display of any alphanumeric field using the X, XX, X*, and X*n* prefix commands. When you use the X prefix command, the type for the field toggles between AN and AX.

When you change the type to AX, the fields are displayed or printed in long hexadecimal format (as opposed to character format).

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Field Selection/Edit                Line 1 of 33

----------- Criteria - Enter 1 or 2 to specify expression by field -----------
1 Id : ____________________________________________________________________ +
2 Sel: (#3='Toyota') & (#7>1997)__________________________________________ +
Offset ____0
Cmd Seq SHE Ref Field Name                              Picture  Type Start Length
               ****  Top of data  ****
___ ___       1 1 VEHICLE-REC                                     AN      1   8475
___ ___       2  2 VEHICLE-TYPE                         X         AX      1      1
___ ___       3  2 VEHICLE-MAKE                         X(20)     AN      2     20
___ ___       4  2 VEHICLE-MODEL                        X(20)     AN     22     20
___ ___       5  2 MODEL-CODE                           X(6)      AN     42      6
___ ___       6  2 VEHICLE-SUB-MODEL                    X(20)     AN     48     20
___ ___       7  2 YEAR-OF-MANUFACTURE                  9(4)      ZD     68      4
___ ___       8  2 NUMBER-ADVERTISED                    9(4)      ZD     72      4
___ ___       9  2 FOR-SALE OCCURS 1 TO 100 TIMES DEPENDING ON
                     NUMBER-ADVERTISED                            AN     76     84
___ ___      10   3 REGO-NUMBER                         X(8)      AN     76      8
Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 62. Field Selection/Edit panel showing VEHICLE-TYPE flagged to display in long hexadecimal mode

Managing templates with the Template Workbench

The Template Workbench is a centralized area in which you can work directly with templates, independently from specific data sets. You can use this area to create, edit or update copybook templates. You cannot create dynamic templates in this area, however, you can edit existing dynamic templates.

Editing a template (E)

To edit a template:

1. In the **Data set name** field in the **Template** section, type the name of the data set in which your existing template is stored.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.
2. If you know it, type the Member name of the template into the relevant field or specify blanks or a pattern for the member name to have a member selection list displayed. You can then select the required member from the list.
3. On the Command line, type the `E` command and press Enter.

   File Manager displays the Field Selection/Edit panel. (See "Field Selection/Edit panel" on page 565 for information about the fields, primary and prefix commands available on that panel.)

Note: If you do not provide a template data set name, but have entered the details of a copybook, File Manager creates a temporary template to edit and use to view data. Before exiting the current panel, you can save the template by specifying a template name and entering the SAVE command. Otherwise the template is discarded.

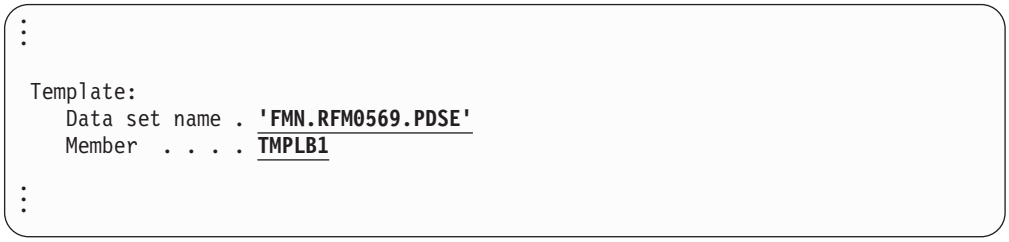
Editing related ID criteria

When you are working with segmented records with related ID criteria (that is, the ID criteria for a segment layout resides in another segment earlier in the physical record), you can edit the related ID criteria in the template as follows:

1. In the **Data set name** field in the **Template** section of the Template Workbench panel, type the name of the data set in which your existing template for segmented data is stored.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.

   If you know it, type the Member name of the template into the relevant field or specify blanks or a pattern for the member name to have a member selection list displayed. You can then select the required member from the list.



2. On the Command line, type the `E` command and press Enter.

File Manager displays the Record Type Selection panel which lists each of the level-01 layouts in the template.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager               Record Type Selection                 Line 1 of 9

Processing Option:    Template for segmented data
Cmd SIE Field Name                                        Prompt   Offset Length
        ****  Top of data  ****
___ S   HEADER-01                                                       0     20
___ S   HEADER-02                                                       0     20
___ S   HEADER-03                                                       0     20
___ S   DETAIL-0101                                                     0     30
___ S   DETAIL-0102                                                     0     30
___ S   DETAIL-0103                                                     0     30
___ S   DETAIL-0201                                                     0     30
___ S   DETAIL-0202                                                     0     30
___ S   DETAIL-0203                                                     0     30
___     ****  End of data  ****




Command ===> ____________________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

3. Select (if currently unselected) the **Template for segmented data** option.
4. Type the E prefix command in the **Cmd** field against the layout for which you want to specify related ID criteria (but which is held in another segment).

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager               Record Type Selection                 Line 1 of 9

Processing Option: /  Template for segmented data
Cmd SIE Field Name                                        Prompt   Offset Length
        ****  Top of data  ****
___ S   HEADER-01                                                       0     20
___ S   HEADER-02                                                       0     20
___ S   HEADER-03                                                       0     20
E   S   DETAIL-0101                                                     0     30
___ S   DETAIL-0102                                                     0     30
___ S   DETAIL-0103                                                     0     30
___ S   DETAIL-0201                                                     0     30
___ S   DETAIL-0202                                                     0     30
___ S   DETAIL-0203                                                     0     30
___     ****  End of data  ****
⋮
```

5. Press Enter.

   File Manager displays the Field Selection/Edit panel showing the selected layout.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 6

---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
0 Rid:                                                                        +
1 Id : ______________________________________________________________________ +
2 Sel: 0_____________________________________________________________________ +
Offset ____0          Enter "/" to OR with related ID
Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                 ****  Top of data  ****
___ ___       1  1 DETAIL-0101                                AN      1     30
___ ___       2   2 SUBTYPE-FIELD                            AN      1      9
___ ___       3    3 SUBTYPECONST                  X(7)      AN      1      7
___ ___       4    3 TYPEREC                       X(2)      AN      8      2
___ ___       5   2 DETAILTEXT01                             AN     10     21
___ ___       6    3 DETAIL01-01                   X(21)     AN     10     21
                 ****  End of data  ****



Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

a. The **0 Rid** line at the top of the panel is blank (unless you have previously specified a related ID expression).

b. The ability to OR the identification criteria with related ID by selecting the processing option.

6. To enter or edit a related ID expression, type `0` (zero) on the command line and press Enter.

   For the selected layout, File Manager displays the RID Selection panel listing the other level-01 layouts in the template.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager          RID Selection for DETAIL-0101              Line 1 of 8

E against layouts to specify related id criteria
Cmd SIE Field Name                                          Prompt   Offset Length
        ****  Top of data  ****
___ SI  HEADER-01                                                         0     20
___ SI  HEADER-02                                                         0     20
___ S   HEADER-03                                                         0     20
___ SI  DETAIL-0102                                                       0     30
___ S   DETAIL-0103                                                       0     30
___ S   DETAIL-0201                                                       0     30
___ S   DETAIL-0202                                                       0     30
___ S   DETAIL-0203                                                       0     30
        ****  End of data  ****




Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

7. Type the E prefix command in the **Cmd** field against the layout in which the related ID criteria are held.

```
:

Cmd SIE Field Name                                          Prompt   Offset Length
        ****  Top of data  ****
E   S   HEADER-01                                                         0     20
___ S   HEADER-02                                                         0     20
___ S   HEADER-03                                                         0     20
___ S   DETAIL-0102                                                       0     30
___ S   DETAIL-0103                                                       0     30
___ S   DETAIL-0201                                                       0     30
___ S   DETAIL-0202                                                       0     30
___ S   DETAIL-0203
:
```

8. Press Enter.

   File Manager displays the Related ID expression panel showing the selected layout.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager    Related Id expression for DETAIL-0101             Line 1 of 7

-------- Criteria - Enter 0 to specify related ID expression by field --------
0 Rid:                                                                        +
Offset     0          Enter "/" to OR with related ID
Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                 ****  Top of data  ****
___ ___       1  1 HEADER-01                                  AN      1     20
___ ___       2   2 ALLGROUP                                  AN      1     20
___ ___       3    3 TYPE-FIELD                               AN      1      8
___ ___       4     4 TYPECONST                    X(6)       AN      1      6
___ ___       5     4 TYPEREC                      X(2)       AN      7      2
___ ___       6    3 HEADERTEXT                               AN      9     12
___ ___       7     4 HEAD01-TEXT                  X(12)      AN      9     12
                 ****  End of data  ****




Command ===>                                                  Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Note:

a. The ability to OR the related criteria with other related ID by selecting the processing option.

9. On the **0 Rid** line at the top of the Related ID expression panel, specify (or edit) the related ID expression. Alternatively, type 0 (zero) on the command line and press Enter to specify the related ID expression by field.

   Typically, the related ID expression specifies the field, and the value it contains, in the layout holding the related ID.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager    Related Id expression for DETAIL-0101             Line 1 of 7

-------- Criteria - Enter 0 to specify related ID expression by field --------
0 Rid: #5='01'                                                                +
Offset     0          Enter "/" to OR with related ID
Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                 ****  Top of data  ****
___ ___       1  1 HEADER-01                                  AN      1     20
___ ___       2   2 ALLGROUP                                  AN      1     20
___ ___       3    3 TYPE-FIELD                               AN      1      8
___ ___       4     4 TYPECONST                    X(6)       AN      1      6
___ ___       5     4 TYPEREC                      X(2)       AN      7      2
___ ___       6    3 HEADERTEXT                               AN      9     12
___ ___       7     4 HEAD01-TEXT                  X(12)      AN      9     12
                 ****  End of data  ****




Command ===>                                                  Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

10. Press the Exit function key (F3) to return to the RID Selection panel.
11. Press the Exit function key (F3) again to return to the Field Selection/Edit panel.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 6

---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
0 Rid: SEG(HEADER-01):#5: TYPEREC    ='01'                                   +
1 Id : ____________________________________________________________________  +
2 Sel: 0___________________________________________________________________  +
Offset ____0          Enter "/" to OR with related ID

                ****  Top of data  ****
___ ___       1  1 DETAIL-0101                                AN      1     30
___ ___       2   2 SUBTYPE-FIELD                            AN      1      9
___ ___       3    3 SUBTYPECONST                  X(7)      AN      1      7
___ ___       4    3 TYPEREC                       X(2)      AN      8      2
___ ___       5   2 DETAILTEXT01                           AN     10     21
___ ___       6    3 DETAIL01-01                  X(21)     AN     10     21
                ****  End of data  ****



Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

You will now see the related ID expression on the **0 Rid** line at the top of the Field Selection/Edit panel.

12. On the **1 Id** and **2 Sel** lines, specify record identification and or selection criteria by field as required.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 6

---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
0 Rid: SEG(HEADER-01):#5: TYPEREC    ='01'                                   +
1 Id : #4='01'_____________________________________________________________  +
2 Sel: 0___________________________________________________________________  +
Offset ____0          Enter "/" to OR with related ID
```

13. **Using SEGNO() and SEGCNT() functions to identify your segments**

    The SEGNO and SEGCNT functions provide a mechanism for checking the current segment number being processed and the number of previous occurrences of a particular segment.

    SEGNO()
    Returns the current segment number.

    SEGCNT('*01 field-name***')**
    Returns the number of previous occurrences of the segment with *01 field name* specified.

The SEGNO and SEGCNT functions can only be specified in a free-format expression

Example of their usage:

Consider the following sample data, where the first segment can be used to determine the number of following A, B, C, or D segments:

```
⋮
       Col 1_____ Insert Length 8160___          Record AT TOP___ Format CHAR
       ----+----10---+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 1234AsegBsegBsegCsegCsegCsegDsegDsegDsegDseg
000002 0202BsegBsegDsegDseg
000003 1232AsegBsegBsegCsegCsegCsegDsegDseg
****** ****  End of data  ****
⋮
```

The data above can be mapped with the following template:

```
⋮
File Manager               Record Type Selection                 Line 1 of 2

Processing Option: _  Template for segmented data
Cmd SIE Field Name                                          Prompt   Offset Length
        ****  Top of data  ****
_e  SI  SEG-HDR                                                         0      4
_e  SI  SEG-A                                                           0      4
_e  SI  SEG-B                                                           0      4
_e  SI  SEG-C                                                           0      4
_e  SI  SEG-D                                                           0      4
        ****  End of data  ****
⋮
```

Examine the criteria for each segment.

Header segment `segno() = 1` ensures its always the first segment on the record.

```
⋮
File Manager                Field Selection/Edit                 Line 1 of 5

---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
0 Rid: ____________________________________________________________________ +
1 Id : segno()=1___________________________________________________________ +
2 Sel: ____________________________________________________________________ +
Offset ____0          Enter "/" to OR with related ID
Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                 ****  Top of data  ****
              1  1 SEG-HDR                                      AN      1      4
              2   2 SEG-NUM-A                        9         ZD      1      1
              3   2 SEG-NUM-B                        9         ZD      2      1
              4   2 SEG-NUM-C                        9         ZD      3      1
              5   2 SEG-NUM-D                        9         ZD      4      1
                 ****  End of data  ****
⋮
```

Note: The segment header contains the number of each of the following segments that occur.

Segment A: note the related expression where we compare the previous occurrences to the value in the header.

```
:
 File Manager                Field Selection/Edit                 Line 1 of 3

 ---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
 0 Rid: SEG(SEG-HDR):segcnt('seg-a') < #2: SEG-NUM-A                          +
 1 Id : ______________________________________________________________________ +
 2 Sel: ______________________________________________________________________ +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
                1  1 SEG-A                                       AN      1      4
                2   2 SEG-TYPE                       X(1)       AN      1      1
                3   2 SEG-DATA                       X(3)       AN      2      3
                   ****  End of data  ****
:
```

```
:
 File Manager          RID Selection for SEG-A                    Line 1 of 4

 E against layouts to specify related id criteria
 Cmd SIE Field Name                                        Prompt   Offset Length
         ****  Top of data  ****
 e__  SI  SEG-HDR                                                      0      4
      SI  SEG-B                                                        0      4
      SI  SEG-C                                                        0      4
      SI  SEG-D                                                        0      4
          ****  End of data  ****
:
```

Press Enter.

```
:
 File Manager    Related Id expression for SEG-A                    Line 1 of 5

 -------- Criteria - Enter 0 to specify related ID expression by field --------
 0 Rid: segcnt('seg-a') < #2                                                  +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
                1  1 SEG-HDR                                     AN      1      4
                2   2 SEG-NUM-A                      9          ZD      1      1
                3   2 SEG-NUM-B                      9          ZD      2      1
                4   2 SEG-NUM-C                      9          ZD      3      1
                5   2 SEG-NUM-D                      9          ZD      4      1
                   ****  End of data  ****
:
```

Note: We use SEGCNT and compare it to the value in the header to ensure that the number of SEG-A identified is determined by the header value.

SEG-B, SEG-C, and SEG-D are all defined with a similar expression as follows:

```
⋮
 File Manager    Related Id expression for SEG-B                  Line 1 of 5

 -------- Criteria - Enter 0 to specify related ID expression by field --------
 0 Rid: segcnt('seg-b') < #3                                                 +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
              1  1 SEG-HDR                                      AN      1      4
              2   2 SEG-NUM-A                         9         ZD      1      1
              3   2 SEG-NUM-B                         9         ZD      2      1
              4   2 SEG-NUM-C                         9         ZD      3      1
              5   2 SEG-NUM-D                         9         ZD      4      1
                  ****  End of data  ****
⋮
```

```
⋮
 File Manager    Related Id expression for SEG-C                  Line 1 of 5

 -------- Criteria - Enter 0 to specify related ID expression by field --------
 0 Rid: segcnt('seg-c') < #4                                                 +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
              1  1 SEG-HDR                                      AN      1      4
              2   2 SEG-NUM-A                         9         ZD      1      1
              3   2 SEG-NUM-B                         9         ZD      2      1
              4   2 SEG-NUM-C                         9         ZD      3      1
              5   2 SEG-NUM-D                         9         ZD      4      1
                  ****  End of data  ****
⋮
```

```
⋮
 File Manager    Related Id expression for SEG-D                  Line 1 of 5

 -------- Criteria - Enter 0 to specify related ID expression by field --------
 0 Rid: segcnt('seg-d') < #5                                                 +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
              1  1 SEG-HDR                                      AN      1      4
              2   2 SEG-NUM-A                         9         ZD      1      1
              3   2 SEG-NUM-B                         9         ZD      2      1
              4   2 SEG-NUM-C                         9         ZD      3      1
              5   2 SEG-NUM-D                         9         ZD      4      1
                  ****  End of data  ****
⋮
```

Viewing the data with a segmented template produces the following where we can see the header segment has been used to determine the number of following segments.

```
⋮
View             FMN.RFM0757.SEGTEST(SEG2)                          Top of 3
                                                      Record AT TOP     Format TABL
       SEG-NUM-A SEG-NUM-B SEG-NUM-C SEG-NUM-D
              #2        #3        #4        #5
         ZD 1:1    ZD 2:1    ZD 3:1    ZD 4:1
             <>        <>        <>        <>
****** ****  Top of data  ****
000001         1         2         3         4
- - -  - - -1- - - SEG-A - - - - - - - - - - - - - - 1 Line(s) suppressed
- - -  - - -1- - - SEG-B - - - - - - - - - - - - - - 2 Line(s) suppressed
- - -  - - -1- - - SEG-C - - - - - - - - - - - - - - 3 Line(s) suppressed
- - -  - - -1- - - SEG-D - - - - - - - - - - - - - - 4 Line(s) suppressed
000002         0         2         0         2
- - -  - - -2- - - SEG-B - - - - - - - - - - - - - - 2 Line(s) suppressed
- - -  - - -2- - - SEG-D - - - - - - - - - - - - - - 2 Line(s) suppressed
000003         1         2         3         2
- - -  - - -3- - - SEG-A - - - - - - - - - - - - - - 1 Line(s) suppressed
- - -  - - -3- - - SEG-B - - - - - - - - - - - - - - 2 Line(s) suppressed
- - -  - - -3- - - SEG-C - - - - - - - - - - - - - - 3 Line(s) suppressed
- - -  - - -3- - - SEG-D - - - - - - - - - - - - - - 2 Line(s) suppressed
⋮
```

How File Manager handles segmented data with related ID criteria

Identification criteria

Identification criteria includes the ability to specify one or more references to fields within other segment types that occur earlier in the physical record in relation to the current segment.

Segment types are checked backwards from the current segment location, so the first matching segment type (for the particular related ID reference) preceding the current segment is the one used for comparison. ID criteria and related ID criteria are ANDed together unless ORing is requested by means of the template option.

Selection criteria

Segment selection criteria affects the selection of the entire record. If a segment within a record fails the selection criteria, the entire record is considered to have failed selection and is not presented or available in any form during subsequent processing.

Template layout selection and deselection

Deselection of a segment type in the template does not affect the processing of the selection criteria. For Edit or View, segments that have been deselected in the template (in order to honor the setting) are always grouped in a not-selected shadow line, regardless of the current grouping setting (the SHOW settings). The shadow line may be hidden or visible, depending on the current settings of the SHADOW primary command.

Unidentified segments

Segments that cannot be identified by explicit ID criteria, or by the implicit criteria of length matching (for those segments that do not have ID criteria specified), are also marked as not-selected segments. However, in Edit or View, these continue to obey the current grouping settings. When in Edit or View and in a multi-line format (TABL, CHAR, HEX or LHEX), and these segments are ungrouped, then the prefix area contains "=ID" to

denote that these segments could not be identified. In SNGL format (as there is no prefix area), a message is issued.

Editor mode of operation
Editing or viewing of segmented data is performed in one of two modes:

In-place, unrestricted size
This mode is used when the template does not contain selection criteria.

In-memory and in-place
In this mode, only as many records as will fit comfortably into the current virtual storage are loaded. Only segments of physical records meeting the template selection criteria are loaded into storage. Only records that are selected are loaded into memory, so this does not restrict the scenario where selection criteria are intended to give a few result records from a large file. This mode is used when the template contains any selection criteria. If you want to examine records and segments occurring later in the file than would be expected to fit into the available storage, you can use the **Start position** field.

Identification and selection currency for Edit and View
Identification and selection criteria are only applied to the records and segments at the time of loading the data into memory (assuming selection criteria have been supplied and this is an in-memory Edit or View) for the editor session. They are not re-assessed at any time during the editor session. For Edit, this means that data changes to segments that potentially affect the segment type, or would have disqualified the record or segment for selection, are not acted on in the current Edit session. To reflect this type of change to the data, it may be saved and the edit session re-entered.

Updating a template from a copybook (U)

If you have made minor changes to the copybook on which a template was based, you can update the template to reflect these changes.

Minor changes that the update process supports include:
- Changing field names without changing field data types
- Changing field data types without changing field names
- Changing the order of fields in a record
- Deleting unreferenced fields
- Inserting new fields
- Changing record length
- Changing the number of occurrences of fields in an array (table)

Major changes that might cause the update process to produce unwanted results include:
- Changing field names and field data types
- Changing field names and the order of fields in a record

If the template contains record identification criteria or record selection criteria, then the update process adjusts the field reference numbers in the criteria to reflect any changes in the field order.

However, if the criteria refer to a field that has been deleted in the copybook, then the update process displays the Field Selection/Edit panel, with references to

deleted fields changed to "#0". Edit the criteria, removing or replacing the obsolete field references, then press the Exit function key (F3) to continue with the update process.

If you change the data type of a field to a dissimilar data type (for example, numeric to non-numeric, or vice versa), then the update process discards any field attributes that you might have defined for that field (such as create attributes).

To update an existing template from a copybook:

Note: File Manager ignores the **Copybook: Data set name** field and instead uses the data set names which were used to create or update the template.

Copybooks can be stored in PDS, PDSE or CA-Panvalet libraries.

1. In the **Data set name** field in the **Template** section, type the name of the data set, in which the template you want updated is stored.

   Templates can be stored in PDS or PDSE data sets but not in a CA-Panvalet library, or a library accessed using the Library Management System Exit.
2. In the **Member** field of the **Templates** section, type the name of your template or specify blanks or a pattern for the member name.
3. On the *Command Line*, type the `U` command and press Enter.

   File Manager takes you to the advanced copybook selection process on the existing source definition for the specified template. When you press the Exit function key (F3), File Manager updates the template. A message is displayed in the upper right corner of the panel, to indicate the success or failure of the process.

You must specify the PDS member name of the copybook you want to use, and the data set name or PDS member name of the template you want to update. The copybook you specify is validated by compiling it with the compiler. For a description of the compilation process, and information about what you can do if errors occur, see "Data description support" on page 134.

Running a function using a copybook or template (RC or RT)

If you entered the Template Workbench from a View or Edit panel (with the TVIEW command), you can use the Template Workbench to edit the current template or to select or create a new template, and then view the data.

To do so:

1. From the View or Edit panel, type `TView` on the Command Line and press Enter.

   The Template Workbench main panel is displayed, with the details of the currently used copybook and/or template in the relevant fields.

   Note: Issuing the TV command followed by a F3 (Exit) or F12 (Cancel) is synonymous to a TV OFF command. You must issue the RC or RT command to resume edit with a template from the workbench.
2. You can edit the current template (`E` command), create a new template from the current copybook (`CC` command), create a new template from the current template (`CM` command) or simply specify a different copybook or template.
3. Make your editing choices and then type the `RC` command (to use the copybook) or `RT` command (to use the template).

The View or Edit panel displays the data, using the newly specified copybook or template.

Note: These two commands are not available when you enter the Template Workbench through option 7.1.

When you enter RC, you are in effect using a temporary template created from the copybook you specify. You cannot edit this template before the panel is run, and the template is not saved. If you want to edit or save the template, use the CC command, and then use the RT command.

To use RC you must specify the PDS member name of the copybook you want to use. The copybook you specify is validated by compiling it with the compiler. For a description of this process, and information about what you can do if errors occur, see "Data description support" on page 134.

If you use the RT command, the template can be either a permanent one previously saved in a PDS member, or a temporary one created for the current panel using the CC command.

Mapping fields (MC or MT)

When you copy data from one data set to another, or compare data in two data sets, you can first provide:

- A template describing the record structure of the input (Copy) or first (Compare) data set
- A template describing the record structure of the output (Copy) or second (Compare) data set

You can then describe the relationship between these record structures (known as field mapping). Field mapping can be created and edited from the Template Workbench, using the MC or MT primary commands, or it can be done "on the fly", within the Copy or Compare Utilities.

Within the Template Workbench and the Copy Utility, the templates are known as the "From" and "To" templates. Within the Compare Utility, they are known as the "Old" and "New" templates. However, the template definitions and field mapping information are independent of any utility and templates intended for use in the Compare Utility can still be created and edited in the Template Workbench.

You specify field mapping information in the "To" (or "New") template. For each field in the "To" template, the field mapping defines which field in the "From" (or "Old") template (if any) are copied to or compared with the field in the "To" template.

Before editing field mapping in a "To" template, you need to specify a "From" template, using either the MC (Map from copybook) or MT (Map from template) command.

To use a copybook to create a new "From" template, specify a copybook on the Template Workbench, then enter the MC command.

To use an existing template as the "From" template, specify an existing template on the Template Workbench, then enter the MT command.

After you enter an MC or MT command, File Manager displays the Map To panel, where you can specify a "To" template, and use the EM or GE commands to begin editing its field mapping (see "Map To panel" on page 595).

Related topics
- "Mapping fields between templates" on page 184
- "Copying data sets" on page 253
- "Comparing data sets" on page 281

Generating and editing field mapping (GM, GE and EM)

The GM (Generate corresponding map) command generates default field mapping between the "From" and "To" templates, replacing any existing field mapping in the "To" template. The GM command does not display the field mapping for editing. For details on the default field mapping generated by File Manager, see "Mapping fields between templates" on page 184.

The EM (Edit mapping) command displays the field mapping for editing. If there is no existing field mapping in the "To" template, then the EM command generates default field mapping (effectively, performing a GM command before displaying the field mapping for editing).

The GE command is similar to EM, except that it always regenerates the default field mapping in the "To" template (replacing any existing field mapping) before displaying the field mapping for editing.

For details on editing the field mapping, see "Mapping fields between templates" on page 184.

Mapping contiguous fields

If you need to map contiguous fields, you may find it more convenient to use the SS, S*n*, or S* prefix commands (see "Field Mapping panel" on page 564).

When you select more than one field at a time, File Manager displays the "From" (or "Old") Field Mapping panel (see "From (or Old) Field Mapping panel" on page 579) for the next selected field each time you press the Exit function key (F3). When you are viewing the last selected field and press the Exit function key (F3), File Manager returns you to the Field Mapping panel (see "Field Mapping panel" on page 564).

Specifying scrambling options

Scrambling data allows you to create test data based on production (or "live") data, but with the ability to change the values of certain fields.

When you use the Copy Utility (option 3.3) to copy data from one data set to another data set, you specify the fields you want scrambled by marking those fields for scrambling in the output template.

You mark a field for scrambling by setting the scrambling options for that field in the template.

You can specify or change scrambling options on the Field Attributes panel. The panel comes in two flavors: one for alphanumeric fields and the other for numeric fields.

- "Selecting a field for specifying scrambling options" on page 181

"Scrambling data" on page 262
"Field Attributes panel - alphanumeric fields" on page 553
"Field Attributes panel - numeric field" on page 559
"Value List Edit panel" on page 705
"Scramble Exit Specification panel" on page 646

Selecting a field for specifying scrambling options

You select a field from the Field Selection/Edit panel (Figure 174 on page 566). To select a field for specifying scrambling options, enter `E` in the **Cmd** field adjacent to the required field. You can enter E against as many fields as you like. For each field you select, File Manager displays a Field Attributes panel.

After you have updated the scrambling options for a field (or any of the other field attributes), when you return to the Field Selection/Edit panel you will see an "E" for that field in the E column (under the "SHE" heading) indicating that the attributes for that field have changed.

The scrambling options you can specify are:
- Scramble type
- Value option
- Value input column
- Value output column
- Minimum and maximum values for a numeric range
- Value data set name

The scrambling options determine how the input data is scrambled (if at all) when it is copied to an output dataset.

Related topics
"Specifying the scramble type"
"Specifying the value option" on page 182
"Specifying value input and out columns" on page 183
"Specifying range values" on page 183
"Specifying a value data set name" on page 183
"Specifying and editing a value list" on page 184

Specifying the scramble type

The **Scramble Type** option determines how data is scrambled when it is copied to the output data set.

Set the scramble type to one of these values:

Blank Data is not scrambled.

1 ("Random")
Data scrambled to produce a random output value on each invocation.

2 ("Repeatable")
Data scrambled to produce the same output value on each invocation.

3 ("Translate")
Data scrambled using input and output values held in the value data set (specified in the **Dsn** field).

This option requires you to select the **Value** option with an input and output column (**In** and **Out**) and value data set name (**Dsn**). All records in the translate data set are loaded into memory for the copy operation. The input and output columns provide the location of the input and output

field values as stored in the data set. Their lengths are determined by the respective input and output fields that have been mapped for the copy operation. If a matching input field value is found, then the corresponding output field value is obtained from the matching record. If no match is found, then the field is initialized to zero, or the value determined by the data create attributes.

4 ("Exit")
Data scrambled using a scrambling exit (as specified on a separate panel). For details about writing a scrambling exit, see the *File Manager Customization Guide*, SC19-4118.

Related topics
"Specifying the value option"
"Specifying value input and out columns" on page 183
"Specifying range values" on page 183
"Specifying a value data set name" on page 183

Specifying the value option

The **Value** option and corresponding fields control the output of the scrambling process.

The way in which you select, or specify, this option differs according to whether the associated field is alphanumeric or numeric:

- For alphanumeric fields, do one of these:
  - Deselect the **Value** option by entering a blank.
  - Select the **Value** option by entering a "/".
- For numeric fields, do one of these:
  - Deselect the **Value** option by entering a blank.
  - Select a range by entering "1".
  - Select the **Value** option by entering "2".

Value option selected ("/")

Selecting this option allows you to provide:

- **A translate data set.** This applies when you select **Scramble Type** 3 ("Translate") with input and output columns and a value data set.
- **A lookup data set.** This applies when you select a **Scramble Type** of 1 ("Random") or 2 ("Repeatable") along with a value data set name. The data set is loaded into memory and the output value is randomly or repeatably selected from one of the loaded records. The output column determines the location of the value that is selected for this field.
- **A value list.** This applies when you select a **Scramble Type** of 1 ("Random") or 2 ("Repeatable") *without* a value data set name and allows you to enter the selection values for this field on the panel displayed when you hit Enter. One of the values you provide is randomly or repeatably selected during a copy operation. The values you enter are stored in the template. Value lists are described in more detail later.

Value option deselected (blank)

Deselecting this option scrambles the input field to produce a random or repeatable output value. **Scramble Type** must be set to 1 ("Random") or 2 ("Repeatable").

Specifying value input and out columns

The input column field (**In**) defines the start location of the input field value on the value data set and is used when the translate process is run during a copy operation to match the input field with a value on the value data set. The length of the field is set to the length of the input field that is mapped to this field during the copy process.

Note: This value is only required when you select **Scramble Type** 3 ("Translate").

The output column field **Out** defines the start location of the output field value on the value data set and is used in these ways during a copy operation:

- **Translate.** If an input field value is matched on the value data set, then the corresponding output value is used.
- **Random** or **Repeatable**. The input value is used to randomly, or repeatably, select an output value from the value data set.

The length of the field is the field length as displayed on the Field Attributes panel.

Note: This value is required when you select **Scramble Type** 3 ("Translate").

If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), and you have also selected the **Value** option, then the output column defaults to 1 if you have specified a value data set name (**Dsn**).

Specifying range values

When you are specifying scramble options for a numeric field, you can specify a range of values from which the resultant scrambled value is to be selected.

You specify the minimum value of the range in the **Min** field and the maximum value of the range in the **Max** field.

You can only select the range option with random and repeatable scrambling. The output value is randomly or repeatably selected from a number within the specified range.

Specifying a value data set name

The output data set name field **Dsn** field defines the value data set. It can be any cataloged sequential, partitioned or VSAM data set containing data that is used to determine the output field value during a copy process.

If you select **Scramble Type** 3 ("Translate"), then the data set must contain the input and output values in the locations provided in the input and output columns.

If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), then the data set must contain the output value in the locations provided in the output column.

Note: This field is required when you select **Scramble Type** 3 ("Translate"). If you select **Scramble Type** 1 ("Random") or 2 ("Repeatable"), and also select the **Value** option, then you can optionally provide a data set name. If you leave this field blank having selected the **Value** option, you are prompted to enter the value list to be stored in the template.

Specifying and editing a value list

A value list is a list of values which is used in conjunction with the scrambling options to select a value with which a field is populated during a copy operation. When the output field is being populated, one of the values in this list is selected. Scrambling options combine the originating value with the random or repeatable seed to select a value from the list.

For random and repeatable scrambling, you can specify a value list inline by leaving the value data set field (**Dsn**) blank and selecting the **Value** option.

To edit the related value list, select the **Value** option and press Enter. File Manager displays the Value List Edit panel.

The values you provide must be valid for the corresponding field type. Deleting a value list turns off the value list option on the previous panel.

To enter hexadecimal values, use the X'hhhh' format:

```
x'C1C2C3'
```

To provide leading blanks or a blank value, use a quoted string:

```
'   SMITH'
'       '
```

Blank value lines (no quotes) are ignored.

Use the FIND and LOCATE commands to bring a value containing or starting with a given string to the current line.

You can use prefix commands to copy, insert, move, repeat or delete lines.

To save the values, press F3. To ignore changes, press Cancel.

Related topics

"Value List Edit panel" on page 705

Mapping fields between templates

Field mapping can be used when copying or comparing data. When copying, you must specify a "From" and a "To" template (see "Copying data sets" on page 253); when comparing, you must specify an "Old" and a "New" template and specify the **Formatted** Compare Option (see "Comparing data sets" on page 281). File Manager uses field mapping information stored in the "To" or "New" template to determine which fields are used in the process.

By default, File Manager generates field mapping information by matching the names of fields in the templates (this is case-sensitive). In many cases, the default field mapping is enough to perform the action you require. However, you can also edit the field mapping, allowing you to specify which record types are mapped and which fields within those record types are mapped. You can set up mapping so that, in the extreme case, a field is mapped to a field with a different name, length, and data type.

Default field mapping

While the same templates can be used for either copying or comparing, you need to understand the way that File Manager uses them in each situation.

First, File Manager pairs record types in the templates by looking for matching field names. Then, within each pair of record types, File Manager maps fields with matching names. The following two examples show the default results of this mapping in the copy and compare processes.

Copying with the default field mapping

A field mapping between "From" and "To" templates could be generated as:

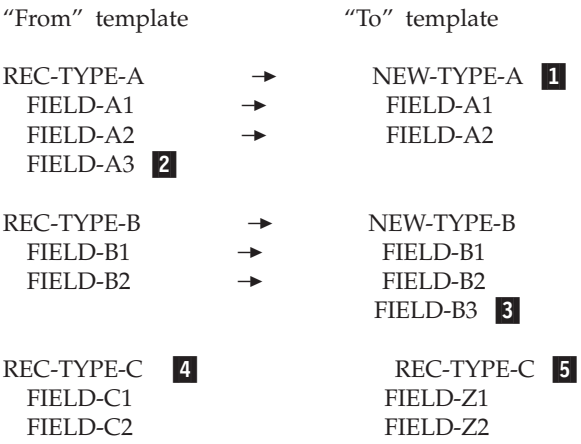


Figure 63. Example of default field mapping

Given an input data set that consists of three records (one of each type defined in the "From" template), like this:

```
├ FIELD-A1 ─┼ FIELD-A2 ─┼ FIELD-A3 ─┤  ← REC-TYPE-A

├ FIELD-B1 ─┼ FIELD-B2 ─┤              ← REC-TYPE-B

├ FIELD-C1 ─┼ FIELD-C2 ─┤              ← REC-TYPE-C
```

then copying the input data set would result in an output data set like this:

```
├ FIELD-A1 ─┼ FIELD-A2 ─┤              ← NEW-TYPE-A 2

├ FIELD-B1 ─┼ FIELD-B2 ─┼ FIELD-B3 ─┤  ← NEW-TYPE-B 3

├ FIELD-C1 ─┼ FIELD-C2 ─┤              ← REC-TYPE-C 4
```

Notes

1 File Manager pairs record types by matching field names within the record types; the names of the record types (level-01 group items, in the copybooks) are not important.

2 There is no matching FIELD-A3 in the "To" template NEW-TYPE-A, so this field is not copied to the output data set.

3 There is no FIELD-B3 in the "From" template to map to this FIELD-B3 in the "To" template. The field is inserted in the output data set; its value is initialized according to the field create attributes (if they exist) in the "To" template, or to zero (if it is a numeric field) or blanks (if it is an alphanumeric field).

4 There is no record type in the "To" template containing fields named FIELD-C1 or FIELD-C2, so the REC-TYPE-C record type in the "From" template is not paired with any record type in the "To" template (even though the "To" template contains a record type named REC-TYPE-C). The

REC-TYPE-C record from the input data set is copied intact to the output data set. To stop this record from being copied, you need to edit the "From" template and deselect the REC-TYPE-C record type.

5 The REC-TYPE-C record type definition in the "To" template is unused, and has no effect on the copy.

Comparing with the default field mapping

A field mapping between "Old" and "New" templates could be generated as:

```
"Old" template              "New" template

OLD-TYPE-A          →       NEW-TYPE-A  1
  FIELD-A1          →         FIELD-A1
  FIELD-A2          →         FIELD-A2
  FIELD-A3  2

OLD-TYPE-B          →       NEW-TYPE-B
  FIELD-B1          →         FIELD-B1
  FIELD-B2          →         FIELD-B2
                              FIELD-B3  3

OLD-TYPE-C                  REC-TYPE-C  4
  FIELD-C1                    FIELD-Z1
  FIELD-C2                    FIELD-Z2
```

Figure 64. Example of default field mapping

With the above field mapping, an "Old" data set containing one record of each type defined in the "Old" template:

```
├─ FIELD-A1 ─┼─ FIELD-A2 ─┼─ FIELD-A3 ─┤  ◄─ OLD-TYPE-A

├─ FIELD-B1 ─┼─ FIELD-B2 ─┤               ◄─ OLD-TYPE-B

├─ FIELD-C1 ─┼─ FIELD-C2 ─┤               ◄─ OLD-TYPE-C
```

and a "New" data set containing one record of each type defined in the "New" template:

```
├─ FIELD-A1 ─┼─ FIELD-A2 ─┤               ◄─ NEW-TYPE-A

├─ FIELD-B1 ─┼─ FIELD-B2 ─┼─ FIELD-B3 ─┤  ◄─ NEW-TYPE-B

├─ FIELD-Z1 ─┼─ FIELD-Z2 ─┤               ◄─ REC-TYPE-C
```

would be compared as follows:

- FIELD-A1 in the NEW-TYPE-A record is compared to FIELD-A1 in the OLD-TYPE-A record.
- FIELD-A2 in the NEW-TYPE-A record is compared to FIELD-A2 in the OLD-TYPE-A record.
- FIELD-A3 in the OLD-TYPE-A record is not compared with any field in any record in the "New" data set. **2**
- FIELD-B1 in the NEW-TYPE-B record is compared to FIELD-B1 in the OLD-TYPE-B record.
- FIELD-B2 in the NEW-TYPE-B record is compared to FIELD-B2 in the OLD-TYPE-B record.
- FIELD-B3 in the NEW-TYPE-B record is not compared with any field in any record in the "Old" data set. **3**

- The REC-TYPE-C records are not compared because there is no mapping match between the fields in the OLD-TYPE-C record and any record in the "New" template. 4

Notes

1 File Manager pairs record types by matching field names within the record types; the names of the record types (level-01 group items, in the copybooks) are not important.

2 There is no matching FIELD-A3 in the “New” template NEW-TYPE-A, so this field is not mapped and is not used in the comparison.

3 There is no FIELD-B3 in the “Old” template to map to this FIELD-B3 in the “New” template, so this field is not mapped and is not used in the comparison.

4 There are no fields named FIELD-C1 or FIELD-C2 in any record type in the “New” template, so the OLD-TYPE-C record type in the “Old” template is not paired with a record type in the “New” template.

The “New” data in the REC-TYPE-C record is reported using the field descriptions in the NEW-TYPE-A record type. This is because the templates do not contain record identification criteria and NEW-TYPE-A is the first record type in the “New” template with a length matching the length of the REC-TYPE-C record in the “New” data set.

The “Old” data is not reported because there is no mapping match between the fields in the OLD-TYPE-C record and any record in the "New" template. Instead, the message " *** Old data is not mapped to any data *** " is reported.

Editing the field mapping

You can edit the field mapping either when performing a copy or a compare, or by using the Template Workbench panel.

To access the field mapping when performing a copy, you must have specified both a “From” and a “To” template and selected the **Edit template mapping** option in the “To” panel (see “Copying using a copybook or template” on page 259).

To access the field mapping when performing a compare, you must have specified both an “Old” and a “New” template in the first two panels and selected the **2. Formatted** Compare Type and **Edit template mapping** option in the third panel (see “Comparing data sets” on page 281).

To edit the field mapping from the Template Workbench panel:

1. On the Template Workbench (option 7) panel, specify a copybook or template that you plan to use for a “From” or “Old” template.

   Although you need to specify this template before you can begin to edit the field mapping, you do not edit the mapping in this template: field mapping information is stored in the template that you specify on the next panel (“To” or “New”).
2. Enter either an MC (Map from copybook) or an MT (Map from template) command.

3. On the Map From panel, specify a copybook (if you want to create a new template) or an existing template that you plan to use for a "To" or "New" template.
4. Enter an EM (Edit mapping) command to edit the field mapping in that template.

Note: Template mapping information is used by both the Compare and Copy Utilities. When using the Template Workbench, the panel labels are those used by the Copy Utility ("From" and "To").

Regardless of the method used to edit the field mapping, one of the following occurs:

- If the "To" template was generated from a copybook containing only one record type or was created as a dynamic template, the Template Mapping panel is displayed. (For panel details, see "Field Mapping panel" on page 564.)
- If the "To" template was generated from a copybook containing more than one record type, the Record Type Mapping panel is displayed. (For panel details, see "Record Type Mapping panel" on page 634).

  Note: The examples in this section assume that you are editing the field mapping from within the Template Workbench.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                    Record Type Mapping

     TO FMN.SFMNSAM1(FMNCCPY2)              FROM FMN.SFMNSAM1(FMNCCPY)
Cmd  Field name                       Len    Field name                       Len
          ****  Top of data  ****
___  NEW-TYPE01                        84 |  REC-TYPE01                        84
___  NEW-TYPE02                       100 |  REC-TYPE02                       100
          ****  End of data  ****









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Describe
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 65. Record Type Mapping panel

To edit the mapping for a particular record type in the "To" template, type an S in the adjacent **Sel** column and press Enter.

File Manager displays the Field Mapping panel, showing the existing mapping between the fields in the "To" template record type and fields in the "From" template.

Setting up a mapping for a field

To set up or change the mapping for a field in the "To" template, enter S against the field name. The From Field Mapping panel is displayed, listing the fields in the "From" template that you can map to the "To" field:

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                    Field Mapping                     Line 1 of 7

To          FMN.SFMNSAM1(FMNCCPY2)
From        FMN.SFMNSAM1(FMNCCPY)

Cmd Lv To Field Name       Type  Len   Lv From Field Name            Type  Len
      ****  Top of data  ****
___   1 NEW-TYPE01          AN    84 |  1 REC-TYPE01                  AN    80
___   2 REC-TYPE            AN     2 |  2 REC-TYPE                   AN     2
___   2 NAME                AN    20 |  2 NAME                        AN    20
___   2 SERIAL-NO           BI     4 |
___   2 AGE                 BI     2 |  2 AGE                         BI     2
___   2 SALARY              PD     4 |  2 SALARY                      PD     4
___   2 MONTH               BI     4 |  2 MONTH                       BI     4
___     ****  End of data  ****




Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Describe
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 66. The Field Mapping panel

In the example panel above, note that there is no “From” field against the “To” field SERIAL-NO. This field is “unmapped”.

To set up or change the mapping for a field in the “To” template, enter S against the field name. The “From” Field Mapping panel is displayed, listing the fields in the “From” template that you can map to the selected “To” field:

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                 From Field Mapping
To   template/copybook : FMN.SFMNSAM1(FMNCCPY2)
From template/copybook : FMN.SFMNSAM1(FMNCCPY)
To field   . . . . . . : #4 SERIAL-NO
From field . . . . . . :
Corresponding(Y/N) . . : N              (Auto map for group items).

Sel   Ref Lvl Old Field                                      Typ Start Length
            ****  Top of data  ****
___     D --  Delete "Old field"
___     1  1  REC-TYPE01                                     AN      1     84
___     2  2  REC-TYPE                                       AN      1      2
___     3  2  NAME                                           AN      3     20
___     4  2  EMPLOYEE-NO                                    BI     23      4
___     5  2  AGE                                            BI     27      2
___     6  2  SALARY                                         PD     29      4
___     7  2  MONTH(13)                                      BI     33      4
            ****  End of data  ****

Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F12=Cancel
```

Figure 67. The From Field Mapping panel

This screen is similar to the Primary Field Selection/Edit panel (Figure 174 on page 566). You edit fields in the same way, by entering E against the field name. For copying in particular, you may wish to edit a field so that you can set the create attributes for the field. These are described in detail in "Changing the attributes for an alphanumeric field" on page 165 and "Changing the attributes for an alphanumeric field" on page 165.

The information at the top of the panel shows you the current mapping. In the panel above, the "To" field is SERIAL-NO. There is currently no "From" field, because there is no "From" field that has the same name.

To specify a "From" field to map to SERIAL-NO, enter S against a "From" field. The information at the top of the panel changes immediately to reflect the new mapping. For example, if you enter S against field #4 (EMPLOYEE-NO), then the information at the top of the panel looks like this:

```
⋮
 File Manager                  From Field Mapping
 To   template/copybook : FMN.SFMNSAM1(FMNCCPY2)
 From template/copybook : FMN.SFMNSAM1(FMNCCPY)
 To  field  . . . . . . : #4 SERIAL-NO
 From field . . . . . . : #4 EMPLOYEE-NO
 Corresponding(Y/N) . . : N               (Auto map for group items).
⋮
```

Press the Exit function key (F3) to return to the Field Mapping panel. This panel now shows that EMPLOYEE-NO is mapped to SERIAL-NO.

```
⋮
File Manager                      Field Mapping                      Line 1 of 7

To          FMN.SFMNSAM1(FMNCCPY2)
From        FMN.SFMNSAM1(FMNCCPY)

Cmd Lv To Field Name       Type  Len   Lv From Field Name             Type  Len
       ****  Top of data  ****
     1 NEW-TYPE01           AN    84 |  1 REC-TYPE01                  AN    80
     2 REC-TYPE             AN     2 |  2 REC-TYPE                   AN     2
     2 NAME                 AN    20 |  2 NAME                        AN    20
     2 SERIAL-NO            BI     4 |  2 EMPLOYEE-NO                 BI     2
     2 AGE                  BI     2 |  2 AGE                         BI     2
     2 SALARY               PD     4 |  2 SALARY                      PD     4
     2 MONTH                BI     4 |  2 MONTH                       BI     4
       ****  End of data  ****
⋮
```

Deleting a field mapping

If the mapping for a field is deleted and the template is used as the "To" template in a copy operation, the value in the output data set is initialized according to the field create attributes in the "To" template, rather than being copied from a field in the input data set. If the template is used as the "New" template in a compare operation, the field is not used in the comparison.

To delete the mapping for a field in a template:

1. Select the field on the Field Mapping panel.
2. On the From Field Mapping panel, enter S next to the `Delete "From field"` line.

Pairing with a different record type

To pair a record type in the "To" template with a different record type in "From" template:

1. Select the level-01 "field name" for the record type on the Field Mapping panel.
2. On the From Field Mapping panel, select the "From" record type that you want to pair with the "To" record type.

Note: This is useful when your "To" template contains only one record type that you wish to map to a second record type in your "From" template.

Mapping group items

In addition to mapping elementary items (as shown in the above example, between the SERIAL-NO and EMPLOYEE-NO fields), you can also map group items (that contain elementary items). If you choose to map a group item in the "From" template without mapping its subsidiary elementary items, then data in the "From" group item is treated as a single alphanumeric field. If you map the "Old" group item to a "To" group item without mapping its subsidiary elementary items, then the data is copied to the output data set (unchanged) or is compared, without regard for any data type differences between the elementary fields in the mapped group items.

If you *do* want to map the elementary items in a "From" group item with the elementary items in a "To" group item then, before selecting the "From" group item on the From Field Mapping panel, type `Y` in the **Corresponding (Y/N)** field. This instructs File Manager to automatically map subsidiary items with the same name in the "From" and "To" group items. (This is just a quick way to set up the mapping for these items; after File Manager has set up these mappings, you can edit them—delete the mapping, or map to a different item—as if you had manually set up the mapping.)

Mapping contiguous fields

If you need to map contiguous fields, you may find it more convenient to use the SS, S*n*, or S* prefix commands (see "Field Mapping panel" on page 564).

When you select more than one field at a time, File Manager displays the From Field Mapping panel (see "From (or Old) Field Mapping panel" on page 579) for the next selected field each time you press the Exit function key (F3). When you are viewing the From Field Mapping panel for the last selected field and press the Exit function key (F3), File Manager returns you to the Field Mapping panel (see "Field Mapping panel" on page 564).

Creating data for a field

If you have specified a "From" field for a copy operation, then the data is transferred to the "To" field, with appropriate truncating or padding. If you have not specified a "From" field, then File Manager creates data for the field.

The rules that apply to the field are exactly the same as those used when creating a new data set. For more information, see "Managing templates" on page 140.

Mapping using different templates

The mapping relationship is stored in the "To" or "New" template. The mapping information is the field identifier and the field name. When you reuse a template to copy or compare data, the mapping information is checked. File Manager makes sure that the field in the specified position has the specified name. If this is not the case, File Manager indicates that the mapping has broken down, with an "Invalid mapping in target" message.

This means that you can use a template that is different from the "From" or "Old" template you used to set up the mapping, provided that the structure of the new "From" or "Old" template does not differ from the structure of the original "From" or "Old" template. That is, no fields that have been used in the mapping are in a different position or have a different name. The new "From" or "Old" template can, however, have different selection conditions.

You can also use a "To" or "New" template as a "From" or "Old" template. However, any mapping information in the template is ignored.

Field mapping rules

File Manager uses the rules shown in Table 3 to determine if a given "from" field can be mapped to a given "to" field. "Yes" indicates that the mapping is valid. "No" indicates that the mapping is invalid.

Table 3. Field mapping rules

"From" field	"To" field		
	Alphanumeric	**Numeric**	**Floating-point**
Alphanumeric[1]	Yes	Yes[2]	Yes[2]
Numeric integer[3]	Yes	Yes	Yes
Numeric non-integer[4]	No	Yes	Yes
Floating-point[5]	No	Yes	Yes

Note:

1. The category "alphanumeric" includes the COBOL categories: alphabetic, alphanumeric, alphanumeric-edited, numeric-edited, and DBCS. PICTURE clause editing characters are ignored. Group fields are treated as elementary alphanumeric fields.
2. The "from" field must consist only of numeric characters, and is treated as a numeric field.
3. The category "numeric integer" includes the following COBOL data types: binary (comp or comp-4 items), packed decimal (comp-3 items), and zoned decimal (PIC 9(*n*) items). The decimal items cannot have an implied decimal point.
4. The category "numeric non-integer" includes packed and zoned decimal items with an implied decimal point and one or more decimal digits.
5. The category "floating point" includes the COBOL floating point data types (comp-1 or comp-2 items). These are representations of real numbers stored with a mantissa and exponent.

These rules describe how a validly mapped "from" field is moved to a "to" field. When the "to" field is:

Alphanumeric

The data is aligned at the leftmost character position and, if necessary, truncated or padded with spaces at the right. The COBOL JUSTIFIED clause, if specified, is ignored.

If the "from" field is a numeric field, the absolute (unsigned) value is used. If the field is defined as signed and the sign occupies a separate character, that character is not moved, and the sending item is considered to be one less character than the actual size.

Numeric

The data is aligned on the assumed decimal point and, if necessary,

truncated or padded with zeros. If an assumed decimal point is not explicitly specified in the field definition, one is assumed immediately to the right of the field.

For a negative number when the "from" field is signed numeric, the data is moved as if both the "from" field and "to" field were described as signed.

When the "from" field is alphanumeric, the data is moved as if the "from" field were described as a signed numeric.

Internal Floating-Point
A decimal point is assumed immediately to the left of the field. The data is aligned then on the leftmost digit position following the decimal point, with the exponent adjusted accordingly.

External Floating-Point
The data is aligned on the leftmost digit position, and the exponent adjusted accordingly.

Copybook View and Print Utility (option 3.13 or 7.2)

You can use the Copybook View and Print Utility (option 3.13 or 7.2) to view or print the field definition information stored in:

- a copybook
- a template that was based on a copybook (not DB2 templates)
- a dynamic template
- an IMS view
- an IMS criteria set

You can also navigate to the Copybook View and Print Utility panel by:

1. Selecting option **7. Templates** from the Primary Option Menu panel to display the Copybook and Template Utility functions panel, and then
2. Selecting option **2. Print** from the Copybook and Template Utility functions panel to display the Copybook View and Print Utility panel.

You can view the report in foreground (view), print the report to your current print data set, or submit a batch job to print the report. When using foreground execution, you can view and print multiple members by selecting them from a member list. When using batch execution, you can specify a generic member name to print multiple members.

To view or print copybook report:

1. From the Utilities Functions menu, select option **13** (Copybook Print and View).
2. Enter the name of the data set that contains your copybook, template, IMS view or IMS criteria set.
3. Enter the name of the member that you wish to view or print, or enter a blank or a pattern to select from a member list.
4. Set your View/Print Option to either **1** (View) or **2** (Print).
5. To run the function in batch (and, if necessary, edit the JCL), select the **Batch Execution** option.
6. To show all occurrences of array elements, select the **Show array elements** option.

7. To show start and end values as hexadecimal offsets, select the **Show start/end as hex offset** option.
8. To show length values as hexadecimal, select the **Show length in hex** option.
9. To show template criteria, select the **Show template criteria** option.

   The information shown when you select the **Show template criteria** option depends on the type of input data set:

Input data set	Information shown
Copybook	The **Show template criteria** option has no effect. Copybook layout reported, with no explicit indication that criteria are not present, and no indication that an extraneous option was selected.
Template with no identification or selection criteria	Template layout reported, with messages to indicate that criteria are not present.
Template with identification or selection criteria	Template layout and criteria reported

10. Press Enter to process the panel.

    If you specified a pattern or a blank for the member name, the Member Selection list is displayed. Enter S in the Sel field for each member that you want to view or print, then press Enter.

    If you selected Batch execution, the generated JCL is displayed. You can tailor this JCL and then submit it for processing.

    Note: Batch execution ignores the View/Print processing option.
11. If you used foreground execution and chose to view the report, the View panel is displayed. You can browse the contents of this panel, or print the contents to the current print data set by entering 1.

Related topics

"Utility Functions menu panel" on page 703
"Template and Copybook Utility functions panel" on page 687
"Copybook View and Print: Entry panel" on page 507

XML templates

A File Manager template can be exported to an external XML format using the Template Export utility. An XML form of the template can be imported into the product using the Template Import utility. The import process can either create or update an existing template using the template definition in its XML format.

You can use an XML version of the template anywhere you can specify a copybook or template.

To use the foreground utility to export a File Manager template, select one of these options:

- Option 7.6 from the File Manager for z/OS Primary Options menu.
- Option 7.2 from the File Manager/DB2 for z/OS Primary Options menu.
- Option 4.8 from the File Manager/IMS for z/OS Primary Options menu.

With the foreground Template Export utility, you can display a member list with all types of templates supported. You can also invoke a template edit against any type of template using the E prefix command from the member list.

The batch utility for exporting a File Manager template is TPEXP.

To use the foreground utility to import a File Manager template, select one of these options:

- Option 7.5 from the File Manager for z/OS primary options menu.
- Option 7.1 from the File Manager/DB2 for z/OS primary options menu.
- Option 4.7 from the File Manager/IMS for z/OS primary options menu.

The batch utility for importing a File Manager template is TPIMP.

Note:

- To import a DB2 template you must be running the import utility from a File Manager/DB2 session, whether foreground or batch. The DB2 object referred to in the template XML must exist in the current subsystem for the File Manager/DB2 session.
- To import an IMS template, view or criteria set you must be running the import utility from a File Manager/IMS session, whether foreground or batch.
- You can export any type of template from any type of File Manager session. The export utility is available from all template menus for convenience.

All aspects of a template can be represented in an XML form. The XML can be created manually by specifying the elements and attributes described in the appendix, or by exporting an existing template. An XML template can be used directly within the product as long as the XML provides all the elements required to create a new template. A good way to determine the XML required to create a template is to use the Template Export Utility to export a similarly defined template.

Sample XML for BASE template

The following XML defines a template that is created from two copybooks, COPY01 and COPY02, that can be found in data set FMN.COPY. The example shows how to provide an 01 field and its name (name01="type01") , and how to specify redefines and range specifications. It also shows how to provide identification criteria.

```
<template lang="COBOL" type="BASE">
 <copybooks>
  <library>FMN.COPY</library>
  <member name="COPY01" name01="type01">
   <redefine level="3"/>
   <sourcerange fromstmt="2"/>
  </member>
  <member name="COPY02" name01="type02">
   <redefine level="3"/>
   <sourcerange fromstmt="2"/>
  </member>
 </copybooks>
 <layout copybook="COPY01">
  <criteria type="ID">
   <exp><![CDATA[#2 = '01']]></exp>
  </criteria>
 </layout>
 <layout copybook="COPY02">
  <criteria type="ID">
```

```
    <exp><![CDATA[#2 = '02']]></exp>
   </criteria>
  </layout>
 </template>
```

Sample XML for IMS template

```
<template lang="COBOL" type="IMS">
 <dbd>DJ1E</dbd>
 <dbdlib>FMIMS.XTEST.DBDLIB</dbdlib>
 <copybooks>
  <library>FMN.IMS.IVP.COPYLIB.COBOL</library>
  <member name="SHIRE" lib="1" segname="SHIRE">
  </member>
  <member name="SHIRENP" lib="1" segname="SHIRENP">
  </member>
  <member name="LINKSUB" lib="1" segname="LINKSUB">
  </member>
  <cobol maxrc="4">
  </cobol>
 </copybooks>
 <layout name="SHIRE" copybook="SHIRE" segment="SHIRE">
  <criteria type="ID">
   <exp><![CDATA[#4 = '1']]></exp>
  </criteria>
 </layout>
 <layout name="SHIRE-TOWN" copybook="SHIRE" segment="SHIRE">
  <criteria type="ID">
   <exp><![CDATA[#4 = '2']]></exp>
  </criteria>
 </layout>
 <layout name="SHIRE-CITY" copybook="SHIRE" segment="SHIRE">
  <criteria type="ID">
   <exp><![CDATA[#4 = '3']]></exp>
  </criteria>
 </layout>
 <layout name="SHIRE-NON-PUBLIC" copybook="SHIRENP" segment="SHIRENP">
 </layout>
 <layout name="SHIRE-SUBURB" copybook="LINKSUB" segment="LINKSUB">
 </layout>
</template>
```

Sample XML for DB2 template

```
<template type="DB2">
 <ssid>DFB2</ssid>
 <db2object>DSN8810.EMP</db2object>
 <db2rel>815</db2rel>
 <layout>
  <criteria>
   <exp><![CDATA[
    WHERE "FIRSTNME" = 'Bob' AND "MIDINIT" = 'C'
    ]]></exp>
  </criteria>
 </layout>
</template>
```

Using the foreground interface to export XML templates

Here is an example of exporting an XML template using the base option 7.6.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager           Template Export Utility
Command ===> ____________________________________________________________________

Template:
   Data set name . 'FMN.SAMPLE.TEMPLATE'
   Member  . . . .          (Blank or pattern for member list)
   Filter  . . . . _________  _________  _________  _________

Export Data set:
   Data set name . 'FMN.XML.EXAMPLE'
   Member mask . . ________

Processing Options:
   Enter "/" to select option            Enter "/" to select option
   _  Batch execution                    /  Replace members
   _  Advanced member selection          7  Copybook and criteria only
   _  Skip member list                   7  Stats On
```

Only copybook and criteria information is exported. This means that field or column data like hold, selection, create or alternate heading are not included. Just the basic definition and any criteria included in the template.

```
 Process   Options   Help
--------------------------------------------------------------------------------
 File Manager            Template Member Selection          Row 00001 of 00013
Command ===> ____________________________________________________ Scroll CSR

Input data set  FMN.SAMPLE.TEMPLATE
Export data set FMN.XML.EXAMPLE

       Name     Prompt   Type Created    Updated              Lang   Ver Descr _
       *        *        *    *          *                    *      *   *
s      ABEND            BASE 2004/02/11 2011/11/28 12:39:09 COBOL    2 This
s      AODA0140         DYN  2011/12/19 2012/09/11 11:32:30 NONE     3
s      CONVT1           DB2  2012/03/02 2012/09/19 09:16:41 NONE     3
       COPY01B          BASE 2012/09/12 2012/09/12 06:23:29 PL/I     3
s      DJ1E             IMS  2008/08/15 2012/08/07 16:53:34 COBOL    3
       DJ1ECR1          CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
s      DJ1ECR2          CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
       DJ1ECR3          CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
s      DJ1EVW           VIEW 2004/02/12 2012/08/03 17:37:53 COBOL    3
       EMPBASE          DB2  2009/03/10 2012/09/10 21:13:15 NONE     3
       EMPDB2           DB2  2009/06/29 2012/09/10 21:11:43 NONE     3
       TEST0102         BASE 2012/01/11 2012/08/07 06:09:56 COBOL    3
       THOGAN           BASE 2004/08/09 2009/05/18 14:25:55 COBOL    2
****  End of data  ****
```

From this member list you can use the `E` or `U` prefix command to edit a template. You can use `EX` or `VX` command to edit or view the XML member of the same name in the export data set.

Press Enter after selecting various templates to be exported.

```
 Process   Options   Help
 File Manager            Template Member Selection          Row 00001 of 00013
Command ===> _____________________________________________________ Scroll CSR

Input data set  FMN.SAMPLE.TEMPLATE
Export data set FMN.XML.EXAMPLE

       Name     Prompt   Type Created    Updated             Lang   Ver Descr _
       *        *        *    *          *                   *      *   *
______ ABEND    *ExpRepl BASE 2004/02/11 2011/11/28 12:39:09 COBOL    2 This
______ AODA0140 *ExpRepl DYN  2011/12/19 2012/09/11 11:32:30 NONE     3
______ CONVT1   *ExpRepl DB2  2012/03/02 2012/09/19 09:16:41 NONE     3
______ COPY01B           BASE 2012/09/12 2012/09/12 06:23:29 PL/I     3
______ DJ1E     *ExpRepl IMS  2008/08/15 2012/08/07 16:53:34 COBOL    3
______ DJ1ECR1           CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
______ DJ1ECR2  *ExpRepl CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
______ DJ1ECR3           CRIT 2008/01/12 2009/06/29 16:57:12 PL/I     3
______ DJ1EVW   *Exportd VIEW 2004/02/12 2012/08/03 17:37:53 COBOL    3
______ EMPBASE           DB2  2009/03/10 2012/09/10 21:13:15 NONE     3
______ EMPDB2            DB2  2009/06/29 2012/09/10 21:11:43 NONE     3
______ TEST0102          BASE 2012/01/11 2012/08/07 06:09:56 COBOL    3
______ THOGAN            BASE 2004/08/09 2009/05/18 14:25:55 COBOL    2
****  End of data  ****
```

You can use the foreground utility to generate the batch JCL to run the export.

Using the foreground interface to import XML templates

Here is an example of importing using the base option 7.5.

```
 Process   Options   Help
File Manager           Template Import Utility
Command ===> ____________________________________________________________

XML Input:
   Data set name . 'FMN.XML.EXAMPLE'_____________________________
   Member  . . . . ________ (Blank or pattern for member list)

Import Template:
   Data set name . 'FMN.IMP.NEW'_________________________________
   Member mask . . ________

Processing Options:
   Enter "/" to select option           Enter "/" to select option
   _  Batch execution                   /  Replace - No update
   _  Advanced member selection         _  Skip member list
```

Press Enter to display the member list.

```
 Process   Options   Help
 File Manager          XML Template Member Selection        Row 00001 of 00009
Command ===> _____________________________________________________ Scroll CSR

XML      data set FMN.XML.EXAMPLE
Template data set FMN.IMP.NEW

       Name     Prompt   Type Created    Updated             Lang   Ver Descr _
       *        *        *    *          *                   *      *   *
______ ABEND             BASE 2013/06/07 2013/06/07 10:29:55 COBOL  XML This
s_____ AODAO140          DYN                                 NONE   XML
______ CONVT1            DB2                                 NONE   XML
______ COPY01B           BASE 2012/12/01 2012/12/01 00:10:19 COBOL  XML
______ DJ1E              IMS  2013/07/03 2013/07/03 13:50:57 COBOL  XML
______ DJ1ECR2           CRIT 2013/05/09 2013/05/10 13:39:57 COBOL  XML
______ DJ1EVW            VIEW 2013/07/03 2013/07/03 13:51:00 COBOL  XML
______ EMPBASE           DB2  2012/09/19 2012/09/19 08:23:12 NONE   XML
s_____ TEST0102          BASE                                COBOL  XML
****  End of data  ****
```

Running under the base product, it is only possible to import dynamic and base templates.

```
 Process   Options   Help
 File Manager          XML Template Member Selection           Template saved
Command ===> _____________________________________________________ Scroll CSR

XML      data set FMN.XML.EXAMPLE
Template data set FMN.IMP.NEW

       Name     Prompt   Type Created    Updated             Lang   Ver Descr _
       *        *        *    *          *                   *      *   *
______ ABEND             BASE 2013/06/07 2013/06/07 10:29:55 COBOL  XML This
______ AODAO140 *ImpRepl DYN                                 NONE   XML
______ CONVT1            DB2                                 NONE   XML
______ COPY01B           BASE 2012/12/01 2012/12/01 00:10:19 COBOL  XML
______ DJ1E              IMS  2013/07/03 2013/07/03 13:50:57 COBOL  XML
______ DJ1ECR2           CRIT 2013/05/09 2013/05/10 13:39:57 COBOL  XML
______ DJ1EVW            VIEW 2013/07/03 2013/07/03 13:51:00 COBOL  XML
______ EMPBASE           DB2  2012/09/19 2012/09/19 08:23:12 NONE   XML
______ TEST0102 *ImpRepl BASE                                COBOL  XML
****  End of data  ****
```

You can use the foreground utility to generate the batch JCL to run the import.

Examples of the batch interface

Example 1. Import - all XML members on FMN.XMLINP to FMN.IMPORT.TEMPLATE

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPIMP DSNIN=FMN.XMLINP,
$$FILEM MEMBER=*,
$$FILEM DSNOUT=FMN.IMPORT.TEMPLATE,
$$FILEM REPLACE=YES
/*
```

Example 2. Import - same as example 1 except using default input and output ddnames. Note keywords INPUT, and OUTPUT can be used to point to different DD names.

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//DDIN  DD DISP=SHR,DSN=FMN.XMLINP
//DDOUT DD DISP=SHR,DSN=FMN.IMPORT.TEMPLATE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPIMP MEMBER=*,REPLACE=YES
/*
```

Example 3. Import from XML that is inline in the jobstream into one member

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPIMP DSNOUT=FMN.TEMPLATE(CBL1),INPUT=*
<template lang="COBOL" type="BASE">
 <copybooks>
  <library>FMN.COBOL</library>
  <member name="COPY01"/>
  <member name="COPY02"/>
 </copybooks>
 <layout copybook="COPY01">
  <criteria type="ID">
   <exp><![CDATA[#2 = '01']]></exp>
  </criteria>
 </layout>
 <layout copybook="COPY02">
  <criteria type="ID">
   <exp><![CDATA[#2 = '02']]></exp>
  </criteria>
 </layout>
</template>
/*
```

Example 4. Export all template members in a PDS - Replace output members, ISPF statistics and only produce XML for copybook and criteria.

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPEXP DSNIN=FMN.IMPORT.TEMPLATE,
$$FILEM MEMBER=*,
$$FILEM DSNOUT=FMN.XMLOUT.NEW,
$$FILEM COPYCRIT=YES,
$$FILEM STATS=YES,
$$FILEM REPLACE=YES
/*
```

Example 5. Export same as example 4 except using default DD names for input and output. Keywords INPUT, and OUTPUT could also be used.

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//SYSPRINT DD SYSOUT=*
//DDIN  DD DISP=SHR,DSN=FMN.IMPORT.TEMPLATE
//DDOUT DD DISP=SHR,DSN=FMN.XMLOUT.NEW
//SYSIN DD *
$$FILEM TPEXP MEMBER=*,
```

```
$$FILEM COPYCRIT=YES,
$$FILEM STATS=YES,
$$FILEM REPLACE=YES
/*
```

Example 6. Export single member to producing XML output on SYSOUT

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB  DD DISP=SHR,DSN=HLQ.SFMNMOD1
//SYSPRINT DD SYSOUT=*
//XMLDD    DD SYSOUT=*
//SYSIN DD *
$$FILEM TPEXP DSNIN=FMN.TEMPLATE,MEMBER=COPY0102,COPYCRIT=YES,
$$FILEM OUTPUT=XMLDD
/*
```

Example 7. Run a compare job using XML dynamic templates specified in the JCL to perform column compares. This compares columns 1 to 5 input file with columns 8 to 12 of the output file , and columns 6 to 10 of the input file with columns 13 to 17 of the output file.

```
//FMBAT EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//DDOLD DD DISP=SHR,DSN=FMN.DATA(COMP1OLD)
//DDNEW DD DISP=SHR,DSN=FMN.DATA(COMP1NEW)
//TDOLD DD DATA
<template type="DYNAMIC">
 <layout>
  <symbol name="comp1" start="1" length="5" type="AN"/>
  <symbol name="comp2" start="6" length="5" type="AN"/>
 </layout>
</template>
/*
//TDNEW DD DATA
<template type="DYNAMIC">
 <layout>
  <symbol name="comp1" start="8"  length="5" type="AN"/>
  <symbol name="comp2" start="13" length="5" type="AN"/>
 </layout>
</template>
/*
//SYSIN DD *
$$FILEM DSM TYPE=FORMATTED,LIST=DELTA
/*
```

Example 8. Print sample file using XML dynamic template - multiple layouts and ID criteria.

```
//FMBAT EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//DDIN  DD DISP=SHR,DSN=hlq.SFMNSAM1(FMNCDATA)
//TDDIN DD DATA
<template type="DYNAMIC">
 <layout name="rec-type01">
  <symbol name="type" start="1" length="2" type="AN"/>
  <symbol name="name" start="3" length="20" type="AN"/>
  <symbol name="empno" start="23" length="2" type="BI"/>
  <symbol name="age"   start="25" length="2" type="BI"/>
  <symbol name="salary"  start="27" length="4" type="PD"/>
  <criteria type="ID">
    <exp><![CDATA[#type = '01']]></exp>
  </criteria>
 </layout>
 <layout name="rec-type02">
  <symbol name="type" start="1" length="2" type="AN"/>
  <symbol name="name" start="3" length="20" type="AN"/>
  <symbol name="title" start="23" length="14" type="AN"/>
```

```
  <symbol name="addr1" start="37" length="20" type="AN"/>
  <symbol name="addr2" start="57" length="20" type="AN"/>
  <symbol name="postcode" start="77" length="4" type="AN"/>
  <criteria type="ID">
    <exp><![CDATA[#type = '02']]></exp>
  </criteria>
 </layout>
</template>
/*
//SYSIN DD *
$$FILEM DSP FORMAT=SNGL
/*
```

Example 9. Run a DB2 import job that either creates or updates a DB2 template specifying new headings for fields. Note we have the XML inline - the input could come from a member.

```
//FMNDB2   EXEC PGM=FMNDB2,PARM=('SSID=ssss,SQID=userid')
//SYSPRINT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
$$FILEM TPIMP DSNOUT=hlq.TEMPLATE(EMPHD),
$$FILEM REPLACE=YES,
$$FILEM INPUT=*
<template type="DB2">
 <db2object>DSN8810.EMP</db2object>
 <layout>
  <symbol name="empno">
   <heading>Employee</heading>
  </symbol>
  <symbol name="FIRSTNME">
   <heading>First Name</heading>
  </symbol>
  <symbol name="MIDINIT">
   <heading>Middle</heading>
  </symbol>
  <symbol name="LASTNAME">
   <heading>Last Name</heading>
  </symbol>
  <symbol name="BIRTHDATE">
   <heading>Birth Date</heading>
  </symbol>
 </layout>
</template>
```

Chapter 5. Filtering records with templates

Once you have created your templates, you can use them to select records within data sets, and then display, copy, print or otherwise manipulate the selected records.

The major tasks and concepts described in this chapter are:

1. "Selecting records with templates"
2. "Defining criteria expressions" on page 207
3. "Filtering record display using templates" on page 224

Selecting records with templates

When working with a data set in a panel or function, you can use a template to select the specific records that you want to use. You can specify two levels of criteria to select the records you want to process.

Record identification criteria
This identifies the unique characteristics of a record type, enabling File Manager to distinguish it from other types of record in the same data set. By default, if you do not specify these criteria, File Manager uses record length to identify a record type.

If the length of a record matches more than one record definition in the template, and no further record identification criteria are specified, File Manager uses the first matching record type in the template to display all records.

Record selection criteria
After a record has been identified as belonging to a particular type, you can narrow down which records of that type you want to process by specifying record selection criteria.

You specify record identification criteria and record selection criteria using comparison expressions. When you specify criteria, you should be aware of the potential for incurring processing overheads as a result of invoking REXX to resolve the criteria expressions.

Related topics
"Selecting records by record type"
"Specifying record identification criteria" on page 204
"Specifying record selection criteria" on page 206

Selecting records by record type

When working with a data set that contains records of more than one record type, you need to use a copybook or copybook template that describes each type. Dynamic templates cannot be used as they cannot specify more than one record type. Provided that the copybook or copybook template contains more than one type description, each time that you opt to edit the template, the Record Type Selection panel displays a list of record types.

Related topics
"Field Selection/Edit panel" on page 565

"Record Type Selection panel" on page 636

Related concepts

"Improving performance using internal expression processing" on page 213

To select records for viewing or editing using record type:

1. On your entry panel, specify a copybook or copybook template that contains more than one record type and edit the template.

   This can be done in a number of ways. For example, from the View Entry, Edit Entry or various Utilities panels, you can select the Edit Template option; on the View or Edit panels, you can enter the `TE` command; or on the Template Workbench panel, you can enter the `E` command.

   The Record Type Selection panel is displayed.

2. On the Record Type Selection panel, type an `S` in the **Cmd** field adjacent to the type you want, and press Enter.

   This toggles the record type between "selected" and "unselected". The command only affects the selection of records where File Manager can identify the record as belonging to a particular type. To do this, you may need to add record identification criteria.

Specifying record identification criteria

Where your copybook or template contains multiple records types, each type is initially distinguished by the record length. Provided that each type has a unique record length, you can select and display the various types. However, if two or more of the record types have the same length, you need to supply some record identification criteria. This usually takes the form of a field that contains a value unique to each record of the same type.

For example, in the following copybook definition, the Orders and Item records are of different lengths. As such, they can be selected and displayed without additional identification criteria.

```
01 ORDERS.
   05 ORDER-ID        PIC X(5).
   05 CUSTOMER-ID     PIC X(5).
   05 ORDER-DATE.
      10 ORDER-YEAR   PIC 9(4).
      10 ORDER-MONTH  PIC 9(2).
      10 ORDER-DAY    PIC 9(2).
01 ITEM.
   05 PRODUCT-ID      PIC X(9).
   05 QUANTITY        PIC 9(4) BINARY.
   05 UNIT-COST       PIC 9(8) BINARY.
```

However, if the records are of identical length, such as in the sample copybook shown below, File Manager is unable to distinguish between these records unless some record identification criteria is used. By examining the data, we can determine that in all REC-TYPE01 records, the REC-TYPE field contains the value '01', and in all REC-TYPE02 records, the REC-TYPE field value is '02'. In this case, both record types include a REC-TYPE field, however, each record type does not need to use the same field in its identification.

```
01 REC-TYPE01.
   05 REC-TYPE                    PIC XX.
   05 NAME                        PIC X(20).
   05 EMPLOYEE-NO                 PIC 9(4) BINARY.
   05 AGE                         PIC 9(4) BINARY.
   05 SALARY                      PIC 9(7) PACKED-DECIMAL.
   05 MONTH                       PIC 9(8) BINARY OCCURS 12 TIMES.
```

```
   05 FILLER                      PIC XX.
01 REC-TYPE02.
   05 REC-TYPE                    PIC XX.
   05 NAME                        PIC X(20).
   05 JOB-TITLE                   PIC X(14).
   05 ADDR1                       PIC X(20).
   05 ADDR2                       PIC X(20).
   05 POSTCODE                    PIC X(4).
```

If any record types in your data have the same record length, it is strongly recommended that you specify record identification criteria to allow File Manager to distinguish between them. Otherwise, the results can be unpredictable.

Even if the length of each record type is unique, it is still recommended that you specify record identification criteria for every record type, to help isolate records that match the length of a valid record type but contain data that is invalid for that type.

To specify record identification criteria:

1. On your entry panel, specify a copybook or copybook template that contains more than one record type and edit the template.

   This can be done in a number of ways. For example, from the View Entry, Edit Entry or various Utilities panels, you can select the Edit Template option; on the View or Edit panels, you can enter the `TE` command; or on the Template Workbench panel, you can enter the `E` command.

   The Record Type Selection panel is displayed.
2. In the **Cmd** field adjacent to the type you want to edit, type the `E` prefix command and press Enter.

   The Field Selection/Edit panel is displayed.
3. Do one of the following:
   - Enter a freeform expression in the **Record Identification Criteria** field.
   - Type 1 on the Command Line and press Enter, then specify your record identification criteria by field.

     The Record Identification Criteria panel is displayed. This panel is identical to the Dynamic Template panel, except that:
     - the **Field Name**, **Start**, **Length** and **Type** fields are not editable;
     - in an 80 x 24 display, the Field Name field is displayed by default (instead of the **Start**, **Length** and **Type** fields);

     When you exit, the expression created is returned to the **Record Identification Criteria** field on the previous panel.

   Note: Freeform criteria expressions and "criteria by field" expressions are mutually exclusive.

Once your records can be identified by type (either by the record length or by record identification criteria), File Manager automatically *suppresses* the display of all but one record type, depending upon the setting of the editor option `Expose (do not group) records of types: Suppressed`. By default, the first record type identified is displayed.

Related topics

"Defining criteria expressions" on page 207
"Displaying "suppressed" records" on page 224

Related concepts
"About expression processing" on page 213

Specifying record selection criteria

After specifying the record identification criteria for each record type, you can specify selection criteria to limit which records from that type you want to work with.

To specify record selection criteria:

1. From your entry panel, edit the template.

   This can be done in a number of ways. For example, from the View Entry, Edit Entry or various Utilities panels, you can select the Edit Template option; on the View or Edit panels, you can enter the `TE` command; or on the Template Workbench panel, you can enter the `E` command.

   If the template contains only one record type, the Field Selection/Edit panel is displayed immediately.

   If the template contains more than one record type, the Record Type Selection panel is displayed. In the **Cmd** field adjacent to the type you want to edit, type the `E` prefix command and press Enter. The Field Selection/Edit panel is displayed.
2. Perform either of these actions
   - Enter a freeform expression in the **Record Selection Criteria** field.
   - Type `2` on the Command Line and press Enter, then specify your record selection criteria by field.

     The Edit Record Selection Criteria panel is displayed. This panel is identical to the Dynamic Template panel, except that:
     - the **Field Name**, **Start**, **Length** and **Type** fields are not editable;
     - in an 80 x 24 display, the Field Name field is displayed by default (instead of the **Start**, **Length** and **Type** fields).

     When you exit, the expression is returned to the **Record Selection Criteria** field on the previous panel.

   Note: Freeform criteria expressions and "criteria by field" expressions are mutually exclusive.

Related topics
"Defining criteria expressions" on page 207
"Displaying "not-selected" records" on page 226

Related concepts
"About expression processing" on page 213

Specifying offset values

When you are editing a template, you can enter an offset value in the Field Selection/Edit panel to change the starting position of the currently displayed record type. If the template contains more than one Level 01 field, you can use the OFFSET primary command in the Record Type Selection panel to change one or more Level 01 fields. (For an explanation of how the offset values work, see "Adjusting your view to allow for header information" on page 231.)

To change the offset in a single record type:

1. From your entry panel, edit the template.

This can be done in a number of ways. For example, from the View Entry, Edit Entry or various Utilities panels, you can select the Edit Template option; on the View or Edit panels, you can enter the TE command; or on the Template Workbench panel, you can enter the E command.

2. If there are multiple record types in your template, enter E in the prefix command line, adjacent to the record type you want to edit.
3. In the Field Selection/Edit panel, enter a positive or negative number, between -32760 and 32760, in the Offset field.

   The offset value is added to the record length of the Level 01 field and to the starting position of all fields within the record type.

To change the offset in multiple record types:

1. From your entry panel, edit the template.

   This can be done in a number of ways. For example, from the View Entry, Edit Entry or various Utilities panels, you can select the Edit Template option; on the View or Edit panels, you can enter the TE command; or on the Template Workbench panel, you can enter the E command.
2. If there are multiple record types in your template, the Record Type Selection panel is displayed.
3. On the Command line, enter the OFFSET primary command and the offset value. For more information on the behavior of the OFFSET primary command in this panel, see "Record Type Selection panel" on page 636.

   By default, the offset value is applied to all record types with a current offset value of zero. To apply the offset value to all record types in the template, regardless of the current offset value, use the ALL keyword with the primary command. To apply the offset value to a specific record type, specify the field name with the primary command.

   The offset value is added to the record length of the Level 01 field and to the starting position of all fields within the specified record type or types.

Defining criteria expressions

This section describes the ways in which you can enter criteria expressions to select records; outlines how the expressions are evaluated and processed within File Manager and provides a list of operators and functions that can be used in criteria expressions.

To specify record identification or record selection criteria, you can enter a freeform expression into a relevant field or you can build the expression "by field". These methods are mutually exclusive. If you create a freeform expression and then create an expression by field, your freeform expression is overwritten. Once you have specified an expression by field, you cannot enter a free form expression until you delete the "by field" expression from the relevant panel.

Related topics

"Entering criteria as a freeform expression" on page 208
"Entering criteria by field" on page 209

Related concepts

"About expression processing" on page 213

Entering criteria as a freeform expression

To enter a freeform expression:

1. Display the Field Selection/Edit panel for your template.
2. In either the **Record Identification Criteria** or **Record Selection Criteria** fields, enter the freeform expression.
3. Press Enter.

   File Manager evaluates the expression to see if it can be processed internally or needs to be passed to REXX. If REXX is required, the message “REXX required” is displayed in the upper right corner. You can choose to either modify the expression so that REXX is not required, or use the expression with REXX processing instead of internal processing.
4. Press Exit (F3) to save your criteria and return to your starting panel.

A freeform expression must take the form of:

```
#ref comp-op value
```

#*ref* Field that returns a value from the current record to be used in the expression. Fields are identified using the # symbol, followed by the field Reference number, for example `#5`.

You can also use the value returned by performing a calculation with a number of fields, for example (#5 + #6). To do this, you must embed the calculation within parentheses and use valid arithmetic operators, together with at least one field reference.

Another alternative is to use a REXX function that lists at least one field as an argument, for example MAX(#5,#6,#7).

comp-op
Comparison Operator. A symbol that expresses the way in which the value derived from the field or field calculation is to be compared with *value*, for example the ">" symbol represents "is greater than". File Manager supports all of the REXX comparison operators.

value Any expression that returns a single value to be used in the criteria expression. This can take the form of a string, a numeric value, a field, a calculation that returns a value or a REXX function.

Logical operators, such as AND (&) and OR (|) can be used to create multiple criteria expressions. Parentheses can be used to control the order of evaluation in calculations and criteria expressions.

Note: At least one field reference must be made in the criteria expression. It is common practice to put this reference on the left side of the equation, although it works equally well on the right side. That is, while `#5 > 100` is the more usual way of expressing this criteria, writing it as `100 <= #5` does not cause an error (note the reversal of the comparison operator). If no field reference is included, the criteria are evaluated as either true or false on their own merit and, when true, all records are returned or, when false, all records become “not-selected”.

Related concepts

“About expression processing” on page 213
“Improving performance using internal expression processing” on page 213
“Overview of REXX expressions” on page 215
“Comparison operators” on page 218
“Arithmetic operators” on page 220
“Logical (boolean) operators” on page 221

"Useful functions" on page 221

Entering criteria by field

By field expressions can be entered on any of these panels:

- Dynamic Template
- Record Identification Criteria
- Record Selection Criteria
- Related ID expression panel

For the purpose of entering criteria "by field", these panels operate in an identical manner.

To enter a single criteria expression by field:

1. Display the appropriate panel for your needs.
2. Type an operator in the **Op** field and a value in the **Value** field.
3. Press Enter.

   File Manager evaluates the expression to see if it can be processed internally or needs to be passed to REXX. If REXX is required, the message "REXX required" is displayed in the upper right corner when you press Enter on the relevant panel. You can choose to either modify the expression so that REXX is not required, or use the expression with REXX processing instead of internal processing.

Note: When exiting from the Record Identification Criteria or Record Selection Criteria panels, your "by field" expression replaces any existing freeform expressions in the relevant field of the Field Selection/Edit panel.

To enter multiple criteria expressions by field:

1. Display the appropriate panel for your needs.
2. Type an operator and value for the each criteria in the appropriate fields.
3. Where you want to use the same field for a number of different criteria, use the available prefix commands to copy and rearrange field definitions within the panel.
4. Use the AND and OR connectors to join the various expressions.
5. Use the **(** and **)** parenthesis fields to group your expressions.
6. Press Enter.

   File Manager evaluates the expression to see if it can be processed internally or needs to be passed to REXX.

For example, using the sample data, you may want to display all the Programmers or Developers in your list. To do this, you would need the Job Title field to appear twice in the panel, each with an operator and value, and you would use the OR connector to join them.

```
⋮
Cmd Con ( Field Name          Op  Value                                    )
*** **** Top of data  ****
___ ____ _ REC-TYPE02         ___ ____________________________________________ _
___ AND  _ REC-TYPE           ___ ____________________________________________ _
___ AND  _ NAME               ___ ____________________________________________ _
___ AND  _ JOB-TITLE          =__ Programmer_________________________________ _
___ OR   _ JOB-TITLE          =__ Developer__________________________________ _
___ AND  _ ADDR1              ___ ____________________________________________ _
⋮
```

"Dynamic Template panel" on page 529
"Record Identification Criteria panel" on page 625
"Record Selection Criteria panel" on page 631
"Related ID expression panel" on page 639

Entering criteria expressions for an array element

You can enter criteria expressions by field for an array element (a data element with an OCCURS clause in COBOL).

You can refer to a subscripted data item on the "by field" display, and optionally enter a subscript, operator, and one or more values. The expression generated uses the TFLD external REXX function.

Alternatively, you can code a free-form expression that refers to any or all the elements of an array by using the TFLD external REXX function.

Figure 68 shows a record layout containing an array element (MONTH-DETAILS).

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Field Selection/Edit                Line 1 of 11

----------- Criteria - Enter 1 to specify expression by field -----------
1 Id : ____________________________________________________________________ +
2 Sel: ____________________________________________________________________ +
Offset      0
Cmd Seq SHE Ref Field Name                          Picture  Type Start Length
                ****  Top of data  ****
___ ___       1 1 RFM0411                                     AN      1    940
___ ___       2  2 REC-TYPE                         X         AN      1      1
___ ___       3  2 FILLER                           X(5)      AN      2      5
___ ___       4  2 CONTRACTOR                       X(40)     AN      7     40
___ ___       5  2 SERIAL-NO                        999999    ZD     47      6
___ ___       6  2 MONTH-DETAILS OCCURS 12 TIMES              AN     53     74
___ ___       7   3 PAY                             S9(8)     ZD     53      8
___ ___       8   3 JOB-LOCATION                    X(20)     AN     61     20
___ ___       9   3 CONTRACT-NO                     999999    ZD     81      6
___ ___      10   3 FIRST-NAME                      X(20)     AN     87     20
___ ___      11   3 SURNAME                         X(20)     AN    107     20
Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 68. Record layout containing an array element

Enter "1" to display the Record Identification Criteria panel shown in Figure 69:

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager           Record Identification Criteria            Line 1 of 11

Cmd Con ( Field Name                            Op  Value                    )
    <-> - <---+----1----+----2----+----3----> <-> <+----1----+----2----+-- -
*** ****  Top of data  ****
___       _ RFM0411                             ___ ____________________________ _
___ AND   _ REC-TYPE                            ___ ____________________________ _
___ AND   _ FILLER                              ___ ____________________________ _
___ AND   _ CONTRACTOR                          ___ ____________________________ _
___ AND   _ SERIAL-NO                           ___ ____________________________ _
___ AND   _ MONTH-DETAILS()                     ___ ____________________________ _
___ AND   _ PAY()                               ___ ____________________________ _
___ AND   _ JOB-LOCATION()                      ___ ____________________________ _
___ AND   _ CONTRACT-NO()                       ___ ____________________________ _
___ AND   _ FIRST-NAME()                        ___ ____________________________ _
___ AND   _ SURNAME()                           ___ ____________________________ _
*** ****  End of data  ****

Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 69. Record Identification Criteria panel

The fields suffixed "()" are dimensioned fields.

Figure 70 shows criteria expressions specified for some of the dimensioned fields within MONTH-DETAILS:

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager           Record Identification Criteria            Line 1 of 11

Cmd Con ( Field Name                            Op  Value                    )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
___       _ RFM0411                             ___ ____________________________ _
___ AND   _ REC-TYPE                            ___ ____________________________ _
___ AND   _ FILLER                              ___ ____________________________ _
___ AND   _ CONTRACTOR                          ___ ____________________________ _
___ AND   _ SERIAL-NO                           ___ ____________________________ _
___ AND   _ MONTH-DETAILS()                     ___ ____________________________ _
___ AND   _ PAY()                               >   (ALL) 5000                  _
___ AND   _ JOB-LOCATION()                      =   (1) 'Kimberly'              _
___ AND   _ CONTRACT-NO()                       ___ ____________________________ _
___ AND   _ FIRST-NAME()                        ___ ____________________________ _
___ AND   _ SURNAME()                           CO  'SMITH', 'JONES'            _
*** ****  End of data  ****

Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 70. Subscripted values specified for dimensioned fields

In Figure 70:

- The default subscript is (ANY) indicating that at least one element must satisfy the condition.
- (ALL) indicates that all elements must satisfy the condition.
- (1) indicates the first element must satisfy the condition.

After entering the criteria as shown in Figure 70 on page 211, pressing F3 causes File Manager to generate the expression:

```
TFLD('#7(ALL)','>',5000) & #8(1) = 'Kimberly' & TFLD('#11','CO','SMITH','JONES')
```

as shown in Figure 71.

Note: The TFLD function is generated (instead of the FLD_CO function) when the CO, or ¬CO operators are specified.

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Field Selection/Edit                Line 1 of 11

----------- Criteria - Enter 1 or 2 to specify expression by field -----------
1 Id : TFLD('#7(ALL)','>',5000) & #8(1) = 'Kimberly' & TFLD('#11','CO','SMI  +
2 Sel: ____________________________________________________________________  +
Offset      0
Cmd Seq SHE Ref Field Name                              Picture  Type Start Length
                ****  Top of data  ****
___ ___       1 1 RFM0411                                        AN      1    940
___ ___       2  2 REC-TYPE                             X        AN      1      1
___ ___       3  2 FILLER                               X(5)     AN      2      5
___ ___       4  2 CONTRACTOR                           X(40)    AN      7     40
___ ___       5  2 SERIAL-NO                            999999   ZD     47      6
___ ___       6  2 MONTH-DETAILS OCCURS 12 TIMES                 AN     53     74
___ ___       7   3 PAY                                 S9(8)    ZD     53      8
___ ___       8   3 JOB-LOCATION                        X(20)    AN     61     20
___ ___       9   3 CONTRACT-NO                         999999   ZD     81      6
___ ___      10   3 FIRST-NAME                          X(20)    AN     87     20
___ ___      11   3 SURNAME                             X(20)    AN    107     20
Command ===> ____________________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 71. TFLD expression generated for fields in an array

Related topics
"TFLD" on page 1152
"Record Identification Criteria panel" on page 625

Related topics
"Dynamic Template panel" on page 529

Related concepts
"About expression processing" on page 213
"Improving performance using internal expression processing" on page 213
"Overview of REXX expressions" on page 215
"Comparison operators" on page 218
"Arithmetic operators" on page 220
"Logical (boolean) operators" on page 221
"Useful functions" on page 221

About expression processing

When an expression is entered (freeform or "by field"), File Manager evaluates the expression to see if it can be processed internally or if it must be passed to REXX.

REXX stands for the REstructured eXtended eXecutor language. REXX is a general purpose programming language, similar to PL/I. It includes extensive parsing capabilities for character manipulation, extensive mathematical capabilities, and numerous built-in functions that perform various processing, searching, and comparison functions.

REXX programs are usually run by a language processor (interpreter). That is, the program is run statement-by-statement, without first being translated to another form (compiled). Because of the interpretive nature of REXX, File Manager is able to exploit the power of REXX when processing record identification and record selection criteria.

Although REXX is a powerful and versatile language, coding selection criteria using REXX is straightforward. You do not need to know how to write REXX programs. All you need to know is how to write REXX comparison expressions. File Manager takes care of turning your selection criteria into a REXX program.

Much of the information in this section has been extracted from the *z/OS TSO/E REXX Reference* and *z/OS TSO/E REXX User's Guide*. For more detailed information about REXX, refer to those manuals.

Improving performance using internal expression processing

For simple criteria, File Manager can process the expression internally without needing to invoke REXX. This can result in faster processing of record identification and record selection criteria. If an expression is not eligible for internal processing, File Manager displays the message "REXX required" when you press Enter on the relevant panel. In this case, you can choose to either modify the expression so that REXX is not required, or use the expression with REXX processing instead of internal processing.

For an expression to be eligible for internal processing, it must consist only of:

- **Literal strings**: These include ordinary quoted strings as well as binary and hexadecimal strings. For a more detailed description of literal strings, see "Overview of REXX expressions" on page 215.
- **Numerics**: Unlike REXX, the internal expression processor distinguishes between character and numeric types. To ensure internal processing uses the correct numeric value, enter numeric data as unquoted strings.

  For example, the internal expression processor treats the string `5` as a number and compares it equal to any other numeric with a value of five. However, it treats the string `'5'` as character data and compares it equal to a string containing X'F5' (with optional leading and trailing blanks for non-strict comparison). REXX treats both strings the same and handles them as character data, but allows them to behave as numbers in appropriate contexts.

  Non-floating point numerics are limited to 31 decimal digits, and floating point numerics are limited to the range of numbers that can be stored as a double-precision floating point number. See "Handling floating point numbers" on page 217 for a more complete discussion of how the internal processor and REXX differ in the way they handle floating point numbers.
- **Comparison operators**: The internal expression processor supports all the REXX comparison operators.

The internal processor performs numeric comparisons when both operands are numeric or contain numeric data (formatted references to numeric fields, or FLD references specifying a numeric formatting type); bit comparisons when both operands are bit fields; otherwise, the internal processor performs character comparisons.

Where both operands have the same value, numeric comparisons match regardless of the length and type of the data. Bit comparisons match only when both bit fields are the same. Leading zero-bits are ignored in the longer field if the fields are not the same length. Character comparisons match only if the two strings are identical, except that leading and trailing blanks are ignored by the non-strict operators. The internal expression processor compares character strings byte by byte according to the standard EBCDIC collating sequence and, where a shorter string matches a longer string up to the end of its length, the longer string is deemed to be greater.

- **Logical operators**: The only logical operators that the internal processor allows are the AND (&), OR (|), exclusive OR (&&) and NOT (¬ or \) operators.
- **Arithmetic prefix operators**: You can use the plus (+) and minus (-) operators before decimal numerics (to indicate sign), and before and within floating point numerics as defined by the REXX syntax.
- **File Manager field references**: The internal expression processor supports both formatted and unformatted references when evaluating identification or selection criteria. Field references are not supported in FASTREXX procedures.
- **Function invocations**: The following functions can be included in an expression that is to be processed internally:
  - FLD
  - FLD_CO
  - FLD_TM
  - FLD_TYPE
  - FLDI
  - FLDO
  - I_LENGTH
  - O_LENGTH
  - PRTCOUNT
  - RECSIN
  - RECSOUT
  - SEGCNT
  - SEGNO
  - TESTC
  - TESTN
  - TFLD
- **Parentheses**

These rules also apply when creating conditions in an IF-THEN-ELSE statement within a FASTREXX procedure.

Related concepts

"Comparison operators" on page 218
216
"Using internal (FASTREXX) processing" on page 412
"FLD" on page 1113

"FLD_CO" on page 1114
"FLD_TM" on page 1118
"FLD_TYPE" on page 1119
"I_LENGTH" on page 1124
"O_LENGTH" on page 1129
"PRTCOUNT" on page 1137
"RECSIN" on page 1140
"RECSOUT" on page 1140
"TFLD" on page 1152

Overview of REXX expressions

REXX expressions consist of one or more *terms* interspersed with zero or more *operators* that denote operations to be carried out on terms. Expressions are evaluated left to right, modified by parentheses and by operator precedence in the usual algebraic manner. When parentheses are encountered (other than those that identify function calls), the entire sub-expression between the parentheses is evaluated immediately when the term is required. Expressions are wholly evaluated, unless an error occurs during evaluation. The REXX language uses a free format. This means you can insert extra spaces between terms without causing an error.

The terms you can use in a selection criteria expression are described below.

Literal strings

A literal string is a sequence including any characters and delimited by the quotation single mark (') or the double quotation mark ("). Use two consecutive double quotation marks ("") to represent a " character within a string delimited by double quotation marks. Use two consecutive quotation marks ('') to represent a ' character within a string delimited by quotation marks. A literal string is a constant and its contents are never modified when it is processed.

These are valid strings:

```
'Fred'
"Don't Panic!"
'You shouldn''t'        /* Same as "You shouldn't" */
```

A string followed immediately by a (is considered to be the name of a function. If followed immediately by the symbol X or x, it is considered to be a hexadecimal string. If followed immediately by the symbol B or b, it is considered to be a binary string.

A hexadecimal string is a literal string, expressed using a hexadecimal notation of its encoding. It is any sequence of zero or more hexadecimal digits (0-9, a-f, A-F), grouped in pairs. A single leading 0 is assumed, if necessary, at the front of the string to make an even number of hexadecimal digits. The groups of digits are optionally separated by one or more blanks, and the whole sequence is delimited by single or double quotation marks, and immediately followed by the symbol X or x. (Neither x nor X can be part of a longer symbol. A hexadecimal string is a literal string formed by packing the hexadecimal digits given. Packing the hexadecimal digits removes blanks and converts each pair of hexadecimal digits into its equivalent character, for example: 'C1'X is the same as the character "A".

Hexadecimal strings let you include characters in a program even if you cannot directly enter the characters themselves. These are valid hexadecimal strings:

```
'ABCD'x
"1d ec f8"X
"1 d8"x
```

A binary string is a literal string, expressed using a binary representation of its encoding. It is any sequence of zero or more binary digits (0 or 1) in groups of 8 (bytes) or 4 (nibbles). The first group may have fewer than four digits; in this case, up to three 0 digits are assumed to the left of the first digit, making a total of four digits. The groups of digits are optionally separated by one or more blanks, and the whole sequence is delimited by matching single or double quotation marks and immediately followed by the symbol b or B.

A binary string is a literal string formed by packing the binary digits given. If the number of binary digits is not a multiple of eight, leading zeros are added on the left to make a multiple of eight before packing. Binary strings allow you to specify characters explicitly, bit by bit.

These are valid binary strings:

```
'11110000'b        /* == 'f0'x                   */
"101 1101"b        /* == '5d'x                   */
'1'b               /* == '00000001'b and '01'x   */
'10000 10101010'b  /* == '0001 0000 1010 1010'b */
''b                /* == ''                      */
```

Symbols

Character strings, without quotation marks, which are translated to uppercase. Any symbol that begins with a # is treated as a reference to a field in the record being processed, and the value of the field is used. All other symbols are treated as constants.

File Manager assigns to each field defined in a template a unique field reference number. When you want to refer to a field in a selection criteria expression, you specify the field's Reference number prefixed by #. You can only refer to fields defined in the record you are currently working with. You cannot refer to fields defined in a different record type. In record identification criteria, you can only refer to fields defined in the static portion of the record (that is, the portion of the record that precedes any variable-length array defined with the OCCURS DEPENDING ON clause).

REXX expression evaluation only processes data in the form of “typeless” character strings (typeless because they are not, as in the case of COBOL, of a particular data type such as binary, packed-decimal, and so forth). Therefore, when you refer to a numeric field, File Manager converts the value of the field to a numeric character string that can be processed by REXX when evaluating the selection criteria expression. The number of integer and decimal digits in the character string is determined by the field definition in your template. For example, if you refer to a packed-decimal field with a COBOL PICTURE clause of 999V99, File Manager converts the value of the field to a character string consisting of numeric digits, a period for the virtual decimal place, and, if the value of the field is negative, a leading sign character (such as -123.45). Note that all numeric fields are treated as signed, regardless of whether the COBOL PICTURE clause contains a sign symbol.

Occasionally, you might want to evaluate the value of a numeric field without converting it to a numeric character string. To do this, prefix the field reference number by #u instead of #. This tells File Manager not to convert the number to a numeric character string. For example, if you

wanted to test a two-byte binary numeric field (with a field reference number of 45) for a special value of X'FFFF', you could code:

```
#u45 == 'FFFF'x
```

When you refer to a field in an array, you must qualify the field reference with the required number of subscripts enclosed in parentheses and separated by commas. The number of subscripts you specify must equal the number of dimensions in the array that contains the field you are referencing. In COBOL terms, there must be a subscript for each OCCURS clause in the hierarchy containing the field, including any OCCURS clause for the field itself. Each subscript you specify must be a positive integer in the range 1 to the maximum number of occurrences of the field as specified in the OCCURS clauses. If the field you refer to is in a variable-length array (specified using the OCCURS DEPENDING ON clause), you should ensure that you do not refer to an occurrence of the field that, for any given record, might not exist. If you do refer to a non-existent field, the selection criteria cannot be satisfied, and the record is not selected.

Note: The object of an OCCURS DEPENDING ON clause must be defined as a field in the static portion of the same record as the array (that is, the portion of the record that precedes any variable-length array defined with the OCCURS DEPENDING ON clause). If the object of the OCCURS DEPENDING ON clause is not so defined, you cannot refer to any fields in the array or any fields in the record that follow the array.

Function call
A call to a REXX built-in function.

Sub-expressions
Terms in an expression bracketed within left and right parentheses.

Related concepts
"Support for variable-length arrays" on page 136

Handling floating point numbers

The internal expression processor operates on internal floating point fields in their native, hexadecimal floating point format. External floating point fields are converted to double-precision internal format before being processed as if they were internal floating point fields. If two fields in a comparison have different precisions, then the field with the greater precision is rounded to the lesser precision, and the comparison is performed using the lesser precision. Because internal floating point fields are converted to external (decimal) format and passed to REXX as strings, and external floating point fields are also passed as strings, the result of evaluating a comparison can differ between REXX and the internal expression processor.

To ensure consistency, pay attention to the following points when coding expressions involving floating point fields:

- When comparing a floating point field to a constant, make sure you use a constant that matches the external representation used by File Manager. For example, `-2.2527494E+08` and `-2.2527495E+08` are both stored as `'C7D6D6C4'x` when entered for a single-precision floating point field, but File Manager always displays the data as -2.2527494E+08. Using either value matches with the internal expression processor, but only -2.2527494E+08 matches with REXX.

- Use hex strings and unformatted field references instead of external format to precisely represent internal floating point values. For example, instead of `#2 = -2.2527494E+08` you can use `#U2 = x2c(C7D6D6C4)`. This technique can only be applied if your expression requires REXX, because the `x2c()` function is not processed internally.
- Comparing floating point fields with different precisions using REXX might not produce the results expected because the external representations of floating point numbers with values that match (with or without rounding) but with different precisions are generally not the same. For example, the external representation of `'C7D6D6C4'x (-2.2527494E+08)` is different from that for `'C7D6D6C4 00000000'x (-2.2527494400000000E+08)` because the extra precision of the field produces a more precise external representation. This situation, where the internal values match exactly can be handled in REXX by truncating the longer field (assuming `#2` refers to a short float and `#3` refers to a long float):

  ```
  #U2 = Substr(#U3,1,4)
  ```

  Simulating the rounding process to achieve the same results as the internal processor, and therefore achieving a more general test of equality between fields of different precision requires the following extraordinary expression:

  ```
  #U2 = d2c(c2d(Substr(#U3,1,4))+Substr(x2b(c2x(Substr(#U3,5,1))),1,1),4)
  ```

  This works by:

  – Truncating the long float to four bytes and converting it to decimal.
  – Adding the high-order byte of the discarded half of the characteristic.
  – Converting the result back to internal format.

  The addition is valid because the characteristic is in the rightmost bits of the field, and any carry from the addition increments the mantissa as required. The only case it does not handle is that of exponent overflow.

Comparison operators

REXX comparison operators compare two terms and return the value 1 if the result of the comparison is true, or 0 otherwise. Comparison operators can compare numbers or character strings. The most commonly used comparison operators are:

Operator	Meaning
=	Equal
¬=, \=	Not equal
<	Less than
¬<, \<	Not less than
>	Greater than
¬>, \>	Not greater than
<=	Less than or equal to
>=	Greater than or equal to
><	Greater than or less than (same as not equal)
<>	Less than or greater than (same as not equal)

The “not” character (¬), is synonymous with the backslash (\). You can use the two characters interchangeably.

When comparing terms using these comparison operators, if both terms in the expression are numeric, REXX performs a numeric comparison. Otherwise, both terms are treated as character strings and REXX performs character comparison. (A number in REXX is a string that contains one or more decimal digits, an optional

decimal point, and an optional leading sign character. The string can contain one or more leading or trailing blanks, and the sign character can be separated from the digits by one or more blanks.)

In a character comparison, leading and trailing blanks are ignored, and the shorter string is padded with blanks on the right. Character comparisons are case-sensitive. Therefore, you should delimit character strings with quotation marks to prevent lowercase characters being translated to upper case. For example, if the field `#4` contains the value `MixedCase` both the following comparison operations would be true:

```
#4 = 'MixedCase'
#4 = '    MixedCase     '
```

but the following comparison operation would not be true:

```
#4 = MixedCase
```

In numeric comparisons, the comparison is effected by subtracting the two numbers (calculating the difference) and then comparing the result with 0. For example, the comparison operation:

```
#6 = 10
```

is identical to the operation:

```
(#6 - 10) = 0
```

In addition to these comparison operators, REXX provides a number of "strict" comparison operators that are mainly intended for use when comparing character strings. The strict comparison operators all have one of the characters defining the operator doubled, such as `==` (strictly equal).

The strict comparison operators are:

Operator	Meaning
==	Strictly equal
¬==, \==	Strictly not equal
<<	Strictly less than
¬<<, \<<	Strictly not less than
>>	Strictly greater than
¬>>, \>>	Strictly not greater than
<<=	Strictly less than or equal to
>>=	Strictly greater than or equal to

When you use the `==` comparison operator (strictly equal), the two character strings being compared must be identical (character by character) and of the same length to be considered strictly equal. Leading and trailing blanks are significant. For example, continuing the example using field `#4` that contains the value `MixedCase`, only the first of the following comparison operations would be true:

```
#4 == 'MixedCase'
#4 == '    MixedCase     '
```

Similarly, the strict comparison operators such as >> or << carry out a simple character-by-character comparison, with no padding of either of the strings being compared. The comparison of the two strings is from left to right. If one string is

shorter than and is a leading substring of another, then it is smaller than (less than) the other. The strict comparison operators do not attempt to perform a numeric comparison on the two terms, and should not be used to compare numeric fields.

Arithmetic operators

You can process numeric terms in comparison expressions using the arithmetic operators:

Operator	Meaning
+	Add
-	Minus
*	Multiply
/	Divide
%	Integer divide (divide and return the integer part of the result)
//	Remainder (divide and return the remainder--not modulo, because the result might be negative)
**	Power (raise a number to a whole-number power)
Prefix -	Same as the subtraction: `0 - number`
Prefix +	Same as the addition: `0 + number`

You can use these operators to produce an intermediate result that you can compare with another term. For example, if the field `#6` contains a numeric value representing an employee's annual salary, and the fields `#15` and `#23` contain numeric values representing the employee's annual travel allowance and annual bonus, respectively, you could use the following comparison to select records for employees with a combined annual payment of greater than $100,000:

```
(#6 + #15 + #23) > 100000
```

For another example, if field `#45` contains the number of sick days an employee is entitled to annually, and the field `#46` contains the number of sick days an employee has used in the current year, you could use the following comparison to select records for employees who have used 50% or more of their sick day entitlements:

```
(#46 / #45) >= .5
```

Note: In each of these examples, the arithmetic sub-expression is contained in parentheses. This ensures that the entire sub-expression is evaluated before the comparison operation.

The order of precedence of arithmetic operators is as follows (highest is at the top):

Operator	Meaning
+ - ¬ \	Prefix operators
**	Power
* /	Multiply and divide
+ -	Add and subtract

For example, * (multiply) has a higher priority than + (add), so `3+2*5` evaluates as `13` (rather than the `25` that would result if strict left to right evaluation occurred). To force the addition to occur before the multiplication, you could rewrite the expression as `(3+2)*5`. Adding the parentheses makes the first three tokens a sub-expression.

Logical (boolean) operators

REXX comparison expressions return a true (1) or false (0) value when processed. Logical operators combine two comparisons and return the true (1) or false (0) value depending on the results of the comparisons.

The logical operators are:

Operator
Meaning

& AND

Returns 1 if both comparisons are true. For example:

```
(4 > 2) & (a = a)        /* true, so result is 1  */
(2 > 4) & (a = a)        /* false, so result is 0 */
```

| Inclusive OR

Returns 1 if at least one comparison is true. For example:

```
(4 > 2) | (5 = 3)            /* at least one is true, so result is 1 */
(2 > 4) | (5 = 3)            /* neither one is true, so result is 0 */
```

&& Exclusive OR

Returns 1 if only one comparison (but not both) is true. For example:

```
(4 > 2) && (5 = 3)   /* only one is true, so result is 1 */
(4 > 2) && (5 = 5)   /* both are true, so result is 0 */
(2 > 4) && (5 = 3)   /* neither one is true, so result is 0 */
```

**Prefix ¬ **
Logical NOT

Returns the opposite response. For example:

```
\ 0                          /* opposite of 0, so result is 1 */
\ (4 > 2)                    /* opposite of true, so result is 0 */
```

Logical expressions are used in complex comparisons and can act as checkpoints to stop unwanted conditions (such as testing a field for a value of zero before using it as a divisor). When you have a series of logical expressions, for clarification, use one or more sets of parentheses to enclose each expression. For example:

```
(#46 = 999) | ((#45 > 0) & (#46 / #45) >= .5)
```

Useful functions

REXX provides a rich set of built-in functions, including character manipulation and conversion functions. Some of these functions might be of use when you are writing your comparison expressions, and they are described below. File Manager also provides some functions that might be of use. They are also described below. To call a function, type the function name directly followed by one or more arguments within parentheses. There can be no space between the function name and the left parenthesis. For example:

```
function(arguments)
```

A function call can contain up to 20 arguments separated by commas. Each argument can be one or more of the following:

Argument
Example
Blank function( )
Constant
function(55)

Symbol
function(#5)
Literal string
function('With a literal string')
Option recognized by function
function(option)
Another function
function(function(arguments))
Combination of argument types
function('Literal string', #5, option)

Some of the built-in functions provided by REXX that you might find useful are:

ABS()

Syntax

Returns the absolute value of a *number*. For example, if you want to select records in which field #12 contains a value in the range -10 to +10, you could specify:

```
ABS(#12) <= 10
```

MAX()

Syntax

Returns the largest number from the list specified. For example, if you want to select records in which any of fields #10, #11, or #12 contains a value greater than 55, you could specify:

```
MAX(#10, #11, #12) > 55
```

MIN()

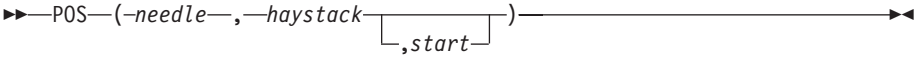
Syntax

Returns the smallest number from the list specified. For example, if you want to select records in which any of fields #10, #11, or #12 contains a value less than 0, you could specify:

```
MIN(#10, #11, #12) < 0
```

POS()

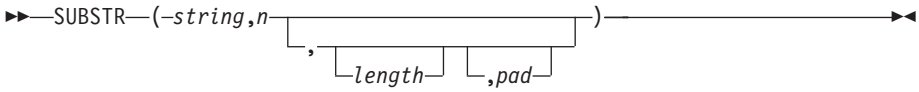
Syntax



Returns the position of one string, *needle,* in another, *haystack*. Returns 0 if *needle* is a null string, or is not found in *haystack*, or if *start* is greater than the length of *haystack*. By default the search starts at the first character of *haystack* (that is, the value of *start* is 1). You can override this by specifying *start* (which must be a positive whole number), the point at which the search starts. For example, if you want to select records in which any character in field #22 is a blank, you could specify: `POS(' ',#22) > 0`.

SUBSTR()

Syntax



Returns the substring of *string* that begins at the *n*th character and is of length *length,* padded with *pad* if necessary. *n* is a positive whole number. If *n* is greater than the length of *string,* then only pad characters are returned.

If you omit *length,* the rest of the string is returned. The default *pad* character is a blank.

For example, if you want to select records in which characters 4-6 of field #22 are the string 'NOT', you could specify:

```
SUBSTR(#22,4,3) == 'NOT'
```

The functions provided by File Manager that you might find useful are shown below.

FLD() Refer to a field from the current input record. For details, see “FLD” on page 1113.

COntains()
Check a field for a list of character values. For details, see “CONTAINS” on page 1111.

NCOntain()
Check a field for a list of numeric values. For details, see “NCONTAIN” on page 1128.

Examples of REXX comparison expressions

The following are examples of REXX comparison expressions:

Example 1
Select records in which the transaction date (field #14) is any date in July 2000, or the transaction value (field #27) is greater than $100,000.00

```
(#14 >= 20000701 & #14 < 20000801) | #27 > 100000.00
```

Example 2
Select records in which the count of credit notes (field #62) is greater than 10% of the count of invoices (field #61)

```
#62 > #61/10
```

Example 3
Select records in which the employee identifier (field #17) starts with any of the letters A, C, or E.

```
CO(SUBSTR(#17,1,1),'A','C','E')
```

Example 4

Select records in which the supplier number (field #23) is 997644 or 997645, and the item description (field #33) contains the word 'CABINET'

```
NCO(#23,997644,997645) & CO(#33,'CABINET')
```

Example 5

Select records in which the 4-byte packed decimal field starting at position 17 in the record contains a negative value

```
FLD(17,4,P) < 0
```

Example 6

Select records in which either the transaction value (field #27) is greater than $50,000.00 and the purchase order number (field #25) starts with characters other than 'TX', or the transaction value is greater than $70,000 and the supplier number (field #23) is 984545, but not if both sets of conditions are true

```
(#27 > 50000.00 & SUBSTR(#25,1,2) ¬== 'TX') &&
(#27 > 70000.00 & #23 = 984545)
```

Note: Examples 1 and 6 would be processed internally.

Filtering record display using templates

One of the main reasons for using a template to view your data is that it allows you to filter the display of your records by applying record identification or record selection criteria.

The tasks described in this section are:

- "Displaying "suppressed" records";
- "Changing display of the record type" on page 225;
- "Displaying "not-selected" records" on page 226;
- "Hiding or showing records in SNGL display format" on page 228;
- "Using the SHOW command in an editor session" on page 228;
- "Viewing segmented data" on page 229 and
- "Adjusting your view to allow for header information" on page 231.

Displaying "suppressed" records

When you are viewing a data set using a template with multiple record types, you can instruct File Manager to include all record types in actions of commands, or only the record type that is current. Only one record type can be "current" at any one time. The inclusion of all record types, or limiting to one record type, is controlled by the SHOW SUP command setting. Once your records can be identified by type (either by the record length or by record identification criteria) and the current SHOW SUP setting is OFF, File Manager *suppresses* the display of all but one record type and groups the other record types into a single display line. By default, the first record type identified is displayed.

You can choose to hide suppressed record groups from display, or represent them by shadow lines:

```
- - - - - record type - - - - - - - - - - - n Line(s) suppressed
```

In TABL display format, you can only show fields from one record type on the screen at a time. Records belonging to other record types are either grouped and re-displayed as suppressed record lines or, if SHOW SUP ON is in effect or the

Expose option Suppressed on the Editor Options panel is selected, represented by a line containing the name of their respective record type. In other display formats, there is no display limitation and the contents of records belonging to other record types are displayed.

Note: Suppressed records are only hidden from display. They are still selected and are still included in any print, copy or create data functions.

You can control whether the suppressed record groups are completely hidden or are represented by shadow lines.

- To hide the shadow line for suppressed record groups, enter `SHADOW SUP OFF` on the Command line The shadow lines disappear, and `SHAD` appears at the top left of the screen, indicating that there are records that are hidden because shadow lines have been turned off.

  You can also hide the shadow line for suppressed records by deselecting the **See shadow lines** option, **Suppressed**, on the Editor Options panel.
- To turn on shadow lines for suppressed record groups, enter `SHADOW SUP ON` on the Command line.

  You can also turn on shadow lines for suppressed records by selecting the **See shadow lines** option, **Suppressed**, on the Editor Options panel.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)               Rec 13 of 40
                                                  Record 13_____ Format TABL
REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY   MONTH(1)
#2       #3                          #4       #5       #6         #7
AN 1:2   AN 3:20                 BI 23:2 BI 25:2  PD 27:4    BI 31:4
<>       <---+----1----+---->     <---+-> <---+-> <---+--> <---+---->
01       Bill McCork                4565      40    76200         10
01       Keith Sampson              2329      40    71550         26
01       John Neptune               3486      40    69000         30
01       Brian Van Der Velde        4574      30    84500         34
01       Ann Norwich                7838      33    67000          2
01       Michael Bevan              5455      46    83250         10
01       Mary Sands                 7790      36    58000         18
01       Antony Burke               7732      46   102000          6
- - - - - - REC-TYPE02  - - - - - - - - - - 20 Line(s) suppressed
****  End of data  ****



Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 72. View panel displaying suppressed records

Related topics

"Managing templates" on page 140
"Defining criteria expressions" on page 207
"SHOW primary command" on page 817
"SHADOW primary command" on page 816

Changing display of the record type

When only one record type is being displayed, to change the current record type in any display format except SNGL:

1. If necessary, turn on the shadow lines for your suppressed records, by entering `SHADOW SUP ON` on the Command line.
2. Type `VIEW` (or `V`) on the Command line.
3. Move the cursor to a shadow line for the suppressed record type you want to view.
4. Press Enter. The suppressed records are displayed and the previous record type becomes suppressed.

Related topics
"Managing templates" on page 140
"Defining criteria expressions" on page 207
"Record Selection Criteria panel" on page 631
"Record Identification Criteria panel" on page 625
"SHADOW primary command" on page 816
"VIEW primary command" on page 828

Displaying "not-selected" records

A record is considered to be *selected* if all of the following conditions are true:

- The record length matches the length of one of the record types (level-01 items) in the template. If any of the data description entries for the fields in a record type included an OCCURS DEPENDING ON clause, then records of various lengths might match a record type; for details, see "Support for variable-length arrays" on page 136.
- The record data matches the record identification criteria for a type with that record length.

  If a record matches the record identification criteria for a record type, but the record length is outside the valid range for that record type, then the record matches the record type only if you choose the **Length error** option on the Editor Options (option 0.6) panel. If you choose this option then such records appear with a `=LGTH` indicator in their prefix area. If you do not choose this option, then the record does not match the record type.
- The record data matches the record selection criteria for that type.
- The record type is indicated as being selected on the Record Selection panel (when editing the template).

Otherwise, a record is considered to be *not-selected* for processing.

You can choose to show not-selected records, hide them, or represent them by shadow lines.

```
- - - - - - - - - - - - - - - - - - - - - - - - n Line(s) not selected
```

- To show not-selected records, enter `SHOW NOT ON` on the Command line. (In SNGL or TABL display format, the record data is formatted according to the structure of the current record type.)

  You can also show not-selected records by selecting the **Expose** option, **Not selected**, on the Editor Options panel.
- To group not-selected records, enter `SHOW NOT OFF` on the Command line. The records are either completely hidden or represented by shadow lines, depending on the current setting of the SHADOW command.
- To turn on shadow lines for not-selected records, enter `SHADOW NOT ON` on the Command line.

  You can also turn on shadow lines for not-selected records by selecting the **See shadow lines** option, **Not selected**, on the Editor Options panel.

- To turn off the shadow lines, completely hiding all but the selected records of the current record type, enter SHADOW NOT OFF on the Command line. The shadow lines disappear, and SHAD appears at the top left of the screen, indicating that there are records that are hidden because shadow lines have been turned off.

  You can also turn off the shadow lines by deselecting the **See shadow lines** option, **Not selected**, on the Editor Options panel.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)                 Rec 0 of 40
                                                  Record 0_______ Format TABL
REC-TYPE NAME                 JOB-TITLE      ADDR1                ADDR2
#2       #3                   #4             #5                   #6         +
AN 1:2   AN 3:20              AN 23:14       AN 37:20             AN 57:20
<>       <---+----1----+----> <---+----1---> <---+----1----+----> <---+----1--
****  Top of data  ****
- - - - - - - - - - - - - - - - - - - - - - - - 20 Line(s) not selected
02       Grant Smith          Developer      22  Montrose St      Thornlie
02       Andrew Apple         Developer      44  Eagle Rise       Riverton
02       Graham Prestcott     Developer      256 Hay St           Cannington
02       Bill Somers          Developer      84  Murchison Rd     Dianella
02       Ted Dexter           Developer      92  Smith St         Belmont
02       Roddy Armstrong      Manager        184 Alexander Dve    Swan View
02       Cliff Roberts        Manager        28  Bern Rd          Middleswan
02       James Browne         Manager        123 Wellington St    Guildford
02       Silvia Carrot        Programmer     48  Small Lane       Mt Pleasant
02       Dan Peters           Programmer     661 Ayton Way        Floreat Park
02       John Laws            Tech Writer    20  Uppercrust Cres  Shents
Command ===> _____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 73. View panel displaying the not-selected shadow line

Related topics

"Managing templates" on page 140
"Defining criteria expressions" on page 207
"SHOW primary command" on page 817
"SHADOW primary command" on page 816

Seeing why a record is not-selected

In an editor session, if a record is not selected, File Manager indicates the reason in the prefix area for the record.

When you use the SHOW command (or one or more of the Expose options on the on the Editor Options panel is selected) to show not-selected records, the prefix area (if displayed) indicates why each record was not selected. One of the following values is displayed in the prefix area of each not-selected record:

=DATA
: Indicates the record was not selected because, although it matched one of the record types identified in the template, it did not match the record selection criteria provided for that record type.

=LGTH
: Indicates the record was not selected because one of the following is true:

  - The record length did not fall within the valid range for any of the record types in the template.

- The record length falls within the valid range for a record type in the template, but that record type contains a variable-length array, and the record length is not consistent with the value of the field that defines the number of items in the array.

Related topics
"Managing templates" on page 140
"Defining criteria expressions" on page 207
"Support for variable-length arrays" on page 136
"SHOW primary command" on page 817
"SHADOW primary command" on page 816

Hiding or showing records in SNGL display format

In SNGL display format, the VIEW command does not apply: the current record type automatically changes to match the record being displayed. The record type used to format the record being displayed is indicated at the top of the screen.

In this display format, shadow lines are not displayed. If SHOW NOT OFF is in effect, then not-selected records are hidden; if SHOW SUP OFF is in effect, then suppressed records are hidden.

If you use the Next function key (F11) or Previous function key (F10) to scroll through records in SNGL display format, then the only indication that a hidden record has been skipped is that the record number (shown in the upper right corner of the screen) skips accordingly.

If you display a not-selected record in SNGL display format, then the record type used to format the data is the record type that was current prior to showing the not-selected record.

Related topics
"Managing templates" on page 140
"Defining criteria expressions" on page 207
"SHOW primary command" on page 817
"SHADOW primary command" on page 816
"VIEW primary command" on page 828
67

Using the SHOW command in an editor session

The SHOW command in an editor session has the following features:

- If you use the SHOW command to show not-selected records, then the prefix area (if displayed) indicates why each record was not selected. For details, see "Seeing why a record is not-selected" on page 227.
- If you use the SHOW command to show not-selected records in TABL display format, and the records contain invalid data when interpreted according to the field definitions for the current record type, then the invalid data is shown as highlighted asterisks. Additionally, in Edit, File Manager moves the cursor to the first occurrence of invalid data.

Examples

In SNGL display format, to show only the selected records for the current record type, enter:

```
SHOW NOT OFF
SHOW SUPP OFF
```

(or just `SHOW ALL OFF`)

In SNGL display format, to show all selected records, enter:

```
SHOW NOT OFF
SHOW SUPP ON
```

(as you step through the records using the Next function key (F11) and the Previous function key (F10), the field headings change to match the type of the current record)

In SNGL display format, to show all records, enter:

```
SHOW NOT ON
SHOW SUPP ON
```

(not-selected records are formatted according to the record type that was current prior to moving to that record).

Related topics

"SHOW primary command" on page 817
"SHADOW primary command" on page 816

Viewing segmented data

When viewing data using a segmented data template, in either the View or Edit panels, only one segment type is displayed at a time in TABL, CHAR, HEX or LHEX display modes, depending upon the current setting of the SHOW SUP command. In this respect, viewing segmented data is very similar to viewing unsegmented data with many record types.

To view a different segment type in the editor panel:

1. Type the `VIEW` (V) primary command on the command line.
2. Place your cursor on the shadow line for the segment type you want to display and press Enter.

To view a different segment type in the Edit panel you can use the method described above or you can:

1. Enter the `V` command in the prefix field for the appropriate shadow line.

Note: Both of these methods are identical to changing record types in unsegmented data.

Because segmented records typically have many segment types, each of which will display as a separate shadow line, you may find it convenient to hide the shadow lines while looking at a particular segment type. To do this:

1. Enter the `SHAD SUP OFF` primary command on the command line.

For example, if this command is applied to the sample data seen in Figure 53 on page 133, the result would be:

```
SHAD
       REC-TYPE DEPT-ID JOB-DESC
       #2       #3      #4
       AN 1:2   AN 3:3  AN 6:20
       <>       <->     <---+----1----+---->
****** ****  Top of data  ****
000001 01       FIN     Accountant
000003 01       FIN     Purchasing Officer
000005 01       FIN     Accounts Receivable
000007 01       MKT     Sales Representative
000009 01       MKT     Promotions Manager
000011 01       MKT     Market Research
000013 01       ADM     Chief Executive
000015 01       ADM     Secretary/PA
000017 01       ADM     Receptionist
000019 01       ADM     Clerical Officer
****** ****  End of data  ****
```

Figure 74. Segmented records with "suppressed" shadow lines turned off.

To switch the segment types, you need to redisplay the shadow lines (SHAD SUP ON), then re-apply the VIEW command to the required Record Type.

Viewing segmented data in SNGL display mode

In SNGL display mode, the File Manager default is to hide the suppressed records. (This is controlled by the **Expose (do not group) records of types** editor option, for which you can select these types of records: **Not selected**, **Suppressed**, **Length error**). This means that when you access SNGL mode, you will only see one segment type as you scroll through the data using the NEXT (F11) and PREVIOUS (F10) commands.

To view all the segment types in SNGL mode, you need to display the suppressed records, by entering the SHOW SUP ON command. When the suppressed records are displayed, the NEXT (F11) and PREVIOUS (F10) commands scroll through each segment type, treating each as a separate record. To scroll through the physical records, you can use the NEXTREC (Shift+F11 or F23) and PREVREC (Shift+F10 or F22) commands. These commands take you to the beginning of the next or previous physical record, that is, you will view the first segment type of each record. You can also use the UP (F7) or DOWN (F8) commands to scroll within the current record.

To find a segment type and then scroll through segmented data in SNGL display mode, viewing the same segment type:

1. Enter the SHOW SUP ON command to display all the segment types.
2. Use the NEXT command (F11) to scroll through the segment types until the type you want is displayed.
3. Enter the SHOW SUP OFF command. This hides the display of the suppressed records.
4. Use the PREV or NEXT commands to scroll, looking at the same segment type each time.

To switch the segment types, you need to redisplay the suppressed records (SHOW SUP ON), scroll to the desired segment type and then re-hide the suppressed records.

Related topics

"NEXTREC primary command" on page 794

"PREVREC primary command" on page 799

Adjusting your view to allow for header information

When you are using a copybook or template to format your records, there are times when the information in your records is displaced because your copybook or template defines its fields as beginning to the right or left of the actual data in the record. For example, your records may include header information, which is not defined in the copybook record structure, or your copybook may include a definition for some header information which is not in the actual record. (PL/I copybooks are sometimes coded with a 4 byte RDW field that File Manager strips from the record, as it isn't normally considered part of the data.) To map your template to your records without changing the originating copybook, you can specify an offset value, which changes the template's field starting positions to the left or right.

The offset value is an integer between -32760 and 32760, which is applied to one or more Level 01 fields in the template. The starting position of each field within the Level 01 record structure is increased or decreased by this value. A positive offset moves the fields to the right and some fields might be mapped beyond the length of the record. Data that falls beyond the length of the record is not represented. You cannot edit a field where the data you supply would fall beyond the physical end of the record (whether the record is of fixed or variable length). A negative offset moves the fields to the left and all fields with a resultant zero or negative start location are removed from the display.

All required fields must have a positive start location. Required fields are fields such as OCCURS DEPENDING ON target fields, PL/I REFER fields, or fields referenced in record identification or record selection criteria. If a required field would have a negative or zero start column value under the supplied offset, the offset is invalid, and thus not allowed. If the first element of a dimensioned field is at a zero or negative start location, that field (all array elements) is removed from the display.

The offset is applied before values are calculated for use in identification or selection criteria.

Related topics

"OFFSET primary command" on page 795
"Managing templates" on page 140
"Selecting records with templates" on page 203
"Defining criteria expressions" on page 207
"Manipulating the display of fields in records" on page 162

Example of a positive offset

If your copybook defined its fields as:

```
Ref Field Name                                  Picture  Type Start    End Length
  1 1 REC-TYPE02                                          AN      1     80     80
  2  2 REC-TYPE                                 XX       AN      1      2      2
  3  2 NAME                                     X(20)    AN      3     22     20
  4  2 JOB-TITLE                                X(14)    AN     23     36     14
  5  2 ADDR1                                    X(20)    AN     37     56     20
  6  2 ADDR2                                    X(20)    AN     57     76     20
  7  2 POSTCODE                                 X(4)     AN     77     80      4
    ****  End of data  ****
```

and your data records, without a template, consisted of (note the header information in bold):

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
000000000102Grant Smith         Developer     22  Montrose St     Thornlie
000000000202Andrew Apple        Developer     44  Eagle Rise      Riverton
000000000302Graham Prestcott    Developer     256 Hay St          Cannington
000000000402Bill Somers         Developer     84  Murchison Rd    Dianella
000000000502Ted Dexter          Developer     92  Smith St        Belmont
000000000602Roddy Armstrong     Manager       184 Alexander Dve   Swan View
```

applying the template generated from this copybook would display your data as:

```
REC-TYPE NAME                JOB-TITLE      ADDR1                ADDR2
#2       #3                  #4             #5                   #6         +
AN 1:2   AN 3:20             AN 23:14       AN 37:20             AN 57:20
<>       <---+----1----+----> <---+----1---> <---+----1----+----> <---+----1--
00       0000000102Grant Smit h         Deve loper     22  Montro se St     Th
00       0000000202Andrew App le        Deve loper     44  Eagle  Rise      Ri
00       0000000302Graham Pre stcott    Deve loper     256 Hay St           Ca
00       0000000402Bill Somer s         Deve loper     84  Murchi son Rd    Di
00       0000000502Ted Dexter           Deve loper     92  Smith  St        Be
00       0000000602Roddy Arms trong     Mana ger       184 Alexan der Dve   Sw
```

By adjusting the template with an offset of 10, the records would display as:

```
REC-TYPE NAME                JOB-TITLE      ADDR1                ADDR2
#2       #3                  #4             #5                   #6         +
AN 11:2  AN 13:20            AN 33:14       AN 47:20             AN 67:20
<>       <---+----1----+----> <---+----1---> <---+----1----+----> <---+----1--
02       Grant Smith         Developer      22  Montrose St      Thornlie
02       Andrew Apple        Developer      44  Eagle Rise       Riverton
02       Graham Prestcott    Developer      256 Hay St           Cannington
02       Bill Somers         Developer      84  Murchison Rd     Dianella
02       Ted Dexter          Developer      92  Smith St         Belmont
02       Roddy Armstrong     Manager        184 Alexander Dve    Swan View
```

Example of a negative offset

If your copybook defined its fields as:

```
Ref Field Name                                 Picture  Type Start    End Length
  1 1 REC-TYPE02                                        AN      1     84     84
  2  2 REC-RDW                                 9(8)     BI      1      4      4
  3  2 REC-TYPE                                XX       AN      5      6      2
  4  2 NAME                                    X(20)    AN      7     26     20
  5  2 JOB-TITLE                               X(14)    AN     27     40     14
  6  2 ADDR1                                   X(20)    AN     41     60     20
  7  2 ADDR2                                   X(20)    AN     61     80     20
  8  2 POSTCODE                                X(4)     AN     81     84      4
    ****  End of data  ****
```

and your data records, without a template, consisted of:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
02Grant Smith         Developer     22  Montrose St     Thornlie            614
02Andrew Apple        Developer     44  Eagle Rise      Riverton            613
02Graham Prestcott    Developer     256 Hay St          Cannington          602
02Bill Somers         Developer     84  Murchison Rd    Dianella            456
02Ted Dexter          Developer     92  Smith St        Belmont             620
02Roddy Armstrong     Manager       184 Alexander Dve   Swan View           613
```

applying the template generated from this copybook would display your data as:

```
REC-RDW REC-TYPE NAME                 JOB-TITLE      ADDR1
         #2 #3       #4                   #5             #6
     BI 1:4 AN 5:2   AN 7:20              AN 27:14       AN 41:20
<---+----1> <>       <---+----1----+----> <---+----1---> <---+----1----+---->
****  Top of data  ****
 -252524647 an       t Smith         Deve loper     22   Montrose St     Thor
 -252526187 dr       ew Apple        Deve loper     44   Eagle Rise      Rive
 -252524647 ah       am Prestcott    Deve loper     256  Hay St          Cann
```

```
 -252525943 ll        Somers         Deve loper     84   Murchison Rd    Dian
 -252517499 d        Dexter          Deve loper     92   Smith St        Belm
 -252520042 dd       y Armstrong     Mana ger       184  Alexander Dve   Swan
```

By adjusting the template with an offset of -4, the records would display as:

```
REC-TYPE NAME                JOB-TITLE      ADDR1                ADDR2
#3       #4                  #5             #6                   #7         +
AN 1:2   AN 3:20             AN 23:14       AN 37:20             AN 57:20
<>       <---+----1----+----> <---+----1---> <---+----1----+----> <---+----1--
****  Top of data  ****
02       Grant Smith         Developer      22  Montrose St      Thornlie
02       Andrew Apple        Developer      44  Eagle Rise       Riverton
02       Graham Prestcott    Developer      256 Hay St           Cannington
02       Bill Somers         Developer      84  Murchison Rd     Dianella
02       Ted Dexter          Developer      92  Smith St         Belmont
02       Roddy Armstrong     Manager        184 Alexander Dve    Swan View
```

Specifying an offset value

You can specify the offset value for a record in three different ways:

1. When you are editing or viewing a data set with a template, use the OFFSET command to enter an offset value. This offset value only applies to the current editor session.

   Note: Under some circumstances, when you use the OFFSET command in an editor session, the command only affects the display formatting of records. You must re-enter the editor session for the select of records or segments to reoccur. These circumstances apply on one of these situations:

   - A template indicating segmented records and some selection criteria are present
   - Record sampling
   - A record limit of MEMORY

   For more information on using the OFFSET primary command, see “OFFSET primary command” on page 795.
2. When you are editing a template, you can enter an offset value in the Field Selection/Edit panel to change the starting position of the currently displayed layout. If the template contains more than one Level 01 field, you can use the OFFSET primary command in the Record Type Selection panel to change one or more Level 01 fields. If you wish, you can save the offset changes in the template, so that they apply to subsequent editor sessions. For more information about specifying an offset in a template, see “Specifying offset values” on page 206.
3. When you are running a File Manager function using a template, batch keywords let you specify offset values. The following functions support the use of offset keywords:

Function	Keywords
DSC	OFFSETIN, OFFSETOUT
DSEB	OFFSETIN
DSM	OFFSETOLD, OFFSETNEW
DSP	OFFSETIN
DSU	OFFSETIN

   For more information about using these keywords, see the relevant function description in Chapter 16, “Functions,” on page 831.

Chapter 6. Managing data sets

Large-scale file management tasks, such as data creation and data copying, can be performed by treating your data sets as entire units, rather than working with individual records within data sets.

The major tasks described in this chapter are:

Similar tasks can be performed on HFS files and between z/OS data sets and HFS files. They are described in Chapter 11, "Using UNIX System Services and the Hierarchical File System," on page 391.

Working with data set lists

The data set list facility provides an easy way to select frequently-used data sets within File Manager and is available for functions whose entry panel allows you to specify the primary data set and, optionally, a copybook or template.

The data set list facility is available with these functions:

- Browse
- Compare
- Copy
- Create
- Edit
- Print
- View

Note: The use of data set lists is not yet supported through the FM/CICS interface.

Data set lists are lists of up to 30 data set names, and optionally the names of the volumes on which the data sets reside, member names, Copybook or Template data set names, and member names.

File Manager maintains a special dynamic "reference list" (called REFLIST) of the last 30 data sets referenced by the function entry panel.

Note: Merely typing a data set name in a panel does not cause File Manager to add that name to the reference list. The data set name has to be actually *allocated* by File Manager before it is added to the reference list.

In addition to File Manager maintaining the reference list, you can create and maintain your own "personal data set lists" with details of up to 30 data sets in each list. You allocate a name to each personal data set list you create so that you can identify it. You can use personal data set lists to retrieve frequently used data

set names and to create customized data set lists. This facility is similar to ISPF personal data set lists, but the lists are not interchangeable between ISPF and File Manager.

Each list (that is, REFLIST and any personal data set lists you have created) includes the data set name and provision for storing associated details such as member name, volume serial, copybook or template data set name, and member name.

File Manager maintains data set lists for each user ID, across functions, and from one File Manager session to another. For example, say you browsed a data set a few days ago and now want to print some records from that same data set, you can find the details of that data set in REFLIST (and any personal data set lists you may have added the data set details to) providing you have not accessed more than 30 data sets in the meantime.

You can open any list (making it the "current data set list") and retrieve details to the function entry panel.

Whenever you view or edit a data set, File Manager records the name of the data set in the reference list (REFLIST). If you specified a volume name on the function entry panel, or a member name for a PDS data set, File Manager also records those details in the reference list. For a PDS, if you do not enter a member name on the function entry panel, but instead select it from the member selection list, File Manager records just the data set name.

File Manager also records the copybook or template data set name and member name, but only if they are *used*. For example, if the Copybook/template processing option is set to **3. None** then, even if the Copybook or Template **Data set name** and **Member** entry fields are not blank, File Manager does not record the contents of those entry fields in the reference list as those details were not used. This avoids the reference list (and potentially any personal data set lists) from holding irrelevant and unnecessary information.

You create personal data set lists by typing in the data set names (and optionally other details), or by saving an existing list (including the reference list) with a new name.

Accessing data set lists

You can access data set lists in one of two ways:

- On the function entry panel, selecting the Process pull-down menu on the action bar
- Using one of the fastpath commands, REFL, REFD, or REFA.

The following sections describe each method of accessing data set lists.

Process pull-down menu

The function entry panels have a pull-down menu (when you select Process on the action bar) that helps you use data set lists. The Process pull-down menu allows you to access data set lists (including the reference list, REFLIST) and define, modify, save and delete personal data set lists. From the Process pull-down menu, you can select one of the following:

Current Data Set List
: Displays the contents of the current data set list and allows you to modify, save, or delete the list.

Personal Data Set Lists
Displays names of all of your personal data set lists, as well as the reference list, REFLIST. You can then open a list to make it active (making it the "current data set list"), and update, delete, or save the list to another name.

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614

Fastpath commands

You can use the following fastpath commands to access and manage data set lists:

REFA Adds the name of the data set (and associated information) most recently referenced by the function entry panel to the specified data set list.

REFD Displays the list of data set lists.

REFL Displays the current data set list.

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614

Managing personal data set lists

You can create, change, or delete data set lists through a set of panels which are accessible by means of the Process pull-down menu available on the function entry panel. You can also access the data set list panels with the REFL and REFD fastpath commands.

Making a list the current data set list

Before you can retrieve information from a data set list, you must make the data set list the *current* data set list.

The current data set list determines:

- The list used by the NRETRIEV primary command (or related function key if allocated to NRETRIEV).
- The list displayed when you select the **Current data set list** option on the Process pull-down menu from the action bar.
- The list displayed when you issue the REFL fastpath command.

To make a data set list the current data set list, perform either of these actions:

1. Enter the REFD fastpath command to display the Personal Data Set Lists panel.
2. Open the required data set list by typing the O line command alongside the name on the list.

1. Save an existing list to a new name. The newly-saved list becomes the current data set list.

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614

Creating a new personal data set list

To create a new personal data set list, perform one of these actions:

- Start with an empty list by selecting the **Personal Data Set Lists** option from the Process pull-down menu on the function entry panel, and use the N (New list) line command to create a new list, then enter the data set names and optional additional information (member name, volume, copybook or template data set name and member name) and use the A (Save as) line command to save it with whatever name you choose.
- Start with the reference list (REFLIST) by selecting the **Personal Data Set Lists** option from the Process pull-down menu on the function entry panel, and use the A (Save as) line command to save it with a new name, and then change the list as needed.
- Start with another personal data set list by selecting the **Personal Data Set Lists** option from the Process pull-down menu on the function entry panel, and use the A (Save as) line command to save it with a new name, and then change the list as needed.

Note:

1. For details about how to add a description to a data set entry, see "Adding a description to an entry in a data set list" on page 239.
2. The N (New list) and A (Save as) line commands are also available when you are modifying an open list.

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614

Editing an existing data set list

You can modify any data set list (including the reference list, REFLIST) by:

- Adding a new data set entry to the list.
- Deleting a data set entry from the list.
- Changing details (including adding a description) for an existing data set entry.

Adding a new entry to a data set list: There are three ways you can add a new entry to a data set list:

- **Overtype an existing entry on the list**

  For an existing entry on the data set list, overtype the details with new information. If necessary, you can also enter new details on any blank lines.
- **Editing the list**
  1. Use the REFL fastpath command or the Process drop-down menu to display the data set list you want to edit on the Personal Data Set List panel.
  2. Enter the E (Extended Edit) line command to display the Edit Personal Data Set List panel.
  3. Type in the details of the new data set entry. When you are adding a new entry to a data set list, you might find the following line commands useful:
     I Insert a list entry
     R Repeat a list entry
- **Using the REFA fastpath command**

  Use the REFA fastpath command to add the name of the data set (and associated information) most recently referenced by the function entry panel to the specified data set list.

  For example:

  ```
  REFA NEWLIST
  ```

adds the name of the data set most recently referenced by the function entry panel to the data set list, NEWLIST.

The list name must begin with an alphabetic character, followed by any valid alphanumeric character for a total length of up to 8 characters. If the specified list name does not exist, File Manager creates it.

Note: For details about how to add a description to a data set entry, see "Adding a description to an entry in a data set list."

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614
"Edit Personal Data Set List panel" on page 543

Deleting an entry from a data set list: To delete an entry in a data set list:

1. Use the REFL fastpath command or the Process drop-down menu to display the data set list you want to edit on the Personal Data Set List panel.
2. Enter the E (Extended Edit) line command to display the Edit Personal Data Set List panel.
3. Type D (Delete a list entry) against the data set entry you want to delete and press Enter. File Manager deletes the entry from the list.

Related topics
"Edit Personal Data Set List panel" on page 543

Changing details for an existing entry in a data set list: To change the details for an existing entry in a data set list:

1. Perform either of these actions:
   - To modify the current data set list, either use the REFL fastpath command or, on the function entry panel select the **Current Data Set List** option from the Process pull-down menu, to open and display the current data set list, and modify the list as required.
   - To modify a data set list other than the current data set list, either use the REFD fastpath command or, on the function entry panel select the **Personal Data Set Lists** option from the Process pull-down menu, to display the list of data set lists and use the O (Open) or E (Edit) line command on the Personal Data Set Lists panel to open and display the required data set list, and modify the list as required.
2. Change any of the existing information by overtyping it.

Note: For details about how to add a description to a data set entry, see "Adding a description to an entry in a data set list."

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614

Adding a description to an entry in a data set list: You can add a description to an individual entry within a data set list. For example, you might want to further identify a data set called EMPOSEAS with the description "Employees who have worked overseas in the last five years".

To add or view descriptions while modifying a personal data set list, on the Personal Data Set panel, perform either of these actions:

- Type the E (Extended Edit) line command to display the Edit Personal Data Set List panel.
- Enter the LISTVIEW command.

File Manager changes the format of the display to include an extra (third) line where you can add or view the description for that data set entry.

Related topics
"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614
"Edit Personal Data Set List panel" on page 543
"LISTVIEW primary command" on page 786

Deleting a data set list

To delete a personal data set list, perform one of these actions:

- Select the **Personal Data Set Lists** option from the Process pull-down menu (or use the REFD fastpath command).
- Place a D next to the list you want to delete and press Enter.
- While viewing a list, place a D in the action field and press Enter.

Note:

1. If you delete the currently active list, then REFLIST becomes the current data set list.
2. You cannot delete the reference list, REFLIST. You can, however, clear its contents. When you use the D line command against REFLIST, File Manager deletes the contents of REFLIST, leaving REFLIST as an empty data set list.

Related topics
"Personal Data Set List panel" on page 612

Retrieving details from a data set list

When you use the function entry panel, you can speed up the selection of commonly-used data sets by retrieving the data set name (and any associated details) from a data set list.

When File Manager retrieves details from a data set list, it fills the **Data set name** entry field on the panel with the retrieved name. If other details (such as the associated member name, volume serial, copybook or template data set name and member name details) are also stored for that entry in the data set list, File Manager also retrieves those details and fills the corresponding entry fields on the panel with the retrieved names.

Note: You can only retrieve details from the current data set list. See "Making a list the current data set list" on page 237.

There are two ways to retrieve data set details:

Progressive retrieval (NRETRIEV primary command)
Progressively retrieves the details of each data set held in the current data set list.

Point-and-shoot retrieval
Retrieves the details of a specific data set from the current data set list.

The following sections describe each of these methods.

Progressive retrieval (NRETRIEV primary command)

To retrieve a data set name from the current data set list:

1. Make the required data set list the current data set list. (See "Making a list the current data set list" on page 237.)
2. From the function entry panel, enter the NRETRIEV primary command (or press the function key to which you have assigned the NRETRIEV primary command).

   File Manager retrieves the first data set name from the current data set list.
3. To retrieve the details for the next data set in the current data set list, repeat the NRETRIEV primary command (or press the associated function key again).
4. Continue to retrieve the details for each data set in the current data set list until you have retrieved the details for the data set you want.

Point-and-shoot retrieval

To retrieve a data set name from the current data set list:

1. Open the required data set list to make it the current data set list. (See "Making a list the current data set list" on page 237.)
2. Place the cursor in the field to the left of the data set name you want, and press Enter.

   The function entry panel is redisplayed with the selected name in the **Data Set Name** field and any associated information in the other entry panel fields.

Related topics

"Personal Data Set List panel" on page 612
"Personal Data Set Lists panel" on page 614
"Edit Personal Data Set List panel" on page 543
"NRETRIEV primary command" on page 794

Allocating program function keys

If you are accessing data set lists frequently, you will probably find it worthwhile to allocate one or more program function (PF) keys to commands used to manage data set lists.

The commands for which you may find it helpful to allocate a PF key are:

LISTVIEW
: On the Personal Data Set List panel, toggles the format of the display to include an extra (third) line where you can add or view the description for that data set entry.

  On the Personal Data Set Lists panel, toggles between the standard and extended list format. The extended list format shows the first 6 data set names for each list.

NRETRIEV
: Retrieves the details of the first data set entry from the current data set list. Repeated use of NRETRIEV returns details of the next data set in the current data set list. You can only use NRETRIEV on the function entry panel.

REFD Displays the list of data set lists.

REFL Displays the current data set list.

To allocate PF keys to one or more of the above commands, use the ISPF command, KEYS.

Creating data sets and records

Using the Data Create Utility, you can:

- Create new records in existing sequential data sets or VSAM data set members, appending to or replacing the existing records.
- Create new records in existing PDS or PDSE members, replacing the existing records.
- Create new records in new VSAM data set members and new PDS or PDSE data set members.
- Create new records in entirely new sequential data sets, VSAM data sets and PDS or PDSE data sets.

With this utility, you can specify the number of records that are to be created and how they are to be initialized. You can use fill characters and patterns to initialize the data. By using a copybook or a template, you can also initialize records at the field level. You can change the data create attributes for individual fields by editing the template.

The tasks described in this section are:

Creating data without using a template

To create data in a new or existing data set, without using a template:

1. From the Primary Option Menu panel, select option **3. Utilities** and then option **1. Create**.

   The Data Create Utility panel is displayed.
2. Do one of the following:
   - Specify the name of an existing data set; for PDS(E) data sets, also specify a new or existing member.

     The new records replace or are appended to the records in the existing data set or member, depending upon how you set the **Disposition** Processing Option.
   - Specify the name of a new data set; for PDS(E) data sets, also specify a new member name.

     When you press Enter to process this panel, you are asked to allocate and define the new data set.
3. If you are creating data in variable length records and you want the new records to have a length less than that of the maximum record size, type a value in the **Record length** field. Otherwise, leave this field empty. (The field is ignored for fixed length and undefined length records.)
4. Enter the number of records that you want to create, in the **Records** field.

   Note: To create an empty data set, specify the number of records as 0.
5. To initialize the data at the record level, enter a value in the **Fillchar** field.
6. If you would like to include a sequence field as part of the record, specify the **Sequence field position**, **Sequence field length** and **Sequence field increment** values.

7. If you have specified the name of a new data set and you want to use another data set as a model, spedify the name of the model data set in the **Like data set** field.
8. Select your **Disposition** Processing Option.

   1. Old/Reuse
   Choose this option when you want your new data to initialize from the beginning of an existing data set, replacing any existing records.

   Note: If your VSAM data set is set to NOREUSE, selecting this option results in an error when the panel is processed.

   2. Mod
   Choose this option when you want your new data to be appended to any existing records in the data set.

   Note: You cannot append records to a PDS or PDSE member or to a packed sequential data set. You also cannot append packed records to a non-packed sequential data set. You must use the Old/Reuse option to replace these records.
9. In the **Copybook or template** processing option, select **3. None**.

   The contents of the Copybook or Template **Data set name** and **Member** fields are ignored.
10. If you want to select a range of PDS(E) members based on one or more criteria, select the **Advanced member selection** option to display the Advanced Member Selection panel.

    For details about selecting a range of PDS(E) members, see "Selecting a range of PDS(E) members" on page 34.
11. If you want to generate the JCL needed to run this function in batch, select the **Batch execution** option. The generated JCL is displayed in an Edit session when the panel is processed.
12. If you want to run a user I/O exit in conjunction with your create action, select the **Use I/O exit** option and then specify the exit name in the adjacent field.

    This option is only available when File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field in the Set System Processing Options panel is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
13. If you want to create the new records in ISPF PACK format, select option **2 Pack** in the ISPF Packing field.

    Note: This option is only applicable to PDS or PDSE members. You cannot specify Pack if you have elected to use an I/O exit.
14. Press Enter.

    If you have specified an existing data set and the process is successful, a message is displayed to show the number of records written to the file.

    If you have specified a new data set name, you are asked to allocate it. When the data set has been successfully allocated, the Data Create panel is displayed again, with a message to show the number of records written to the file.

Related topics

"Specifying a data set and a member name" on page 16
"Allocating a new data set" on page 246
"Using File Manager functions in batch jobs" on page 401

"Data Create Utility panel" on page 515
"DSG (Data Set Generate)" on page 922

Creating data sets using a template

When you use a template to create a data set, the template provides structure for the data. You can also use "create attributes" in the template to initialize field values in the data set.

Initialization can be performed at two levels: record level and field level. Field-level initialization takes place when you specify that a copybook or template is to be used. In this case, the record is first filled with the character specified by the **Fillchar** entry field. Selected fields are then individually initialized using the field create attributes specified in a template.

When File Manager is creating a data set, it uses only the first record type described in the template. The length of the records written depends upon whether the records are fixed-length or variable-length, and the length of the record description in the template:

- For fixed-length records, all records are created with the record length specified for the data set. If the length of the record in the template is less than the data set record length, the records are padded with the value specified for the **Fillchar** field.
- For variable-length records, the length of each record depends upon the length of the record built using the field attributes in the template. If the template record description contains one or more variable-length arrays, the length of the record varies according to the values assigned to the objects of the OCCURS DEPENDING ON clauses or, in the case of a template created from a PL/I copybook, arrays or variables whose number of elements or length is controlled by a REFER variable.

Note: The length of the record in the template must always be less than or equal to the maximum data set record length.

To create data in a new or existing data set, using a template:

1. From the Primary Option Menu panel, select option **3. Utilities** and then option **1. Create**.

   The Data Create Utility panel is displayed.
2. Do one of the following:
   - Specify the name of an existing data set; for PDS(E) data sets, also specify a new or existing member.

     The new records replace or are appended to the records in the existing data set or member, depending upon how you set the **Disposition** Processing Option.
   - Specify the name of a new data set; for PDS(E) data sets, also specify a new member name.

     When you press Enter to process this panel, you are asked to allocate and define the new data set.
3. If you are creating data in variable length records and you want the new records to have a length less than that of the maximum record size, type a value in the **Record length** field. Otherwise, leave this field empty. (The field is ignored for fixed length and undefined length records.)
4. Enter the number of records that you want to create, in the **Records** field.

Note: To create an empty data set, specify the number of records as 0.

5. To initialize the data at the record level, enter a value in the **Fillchar** field.

   To initialize data at the field level, you must use a template in which the Create Attributes values have been defined for the template fields. If these values have not yet been defined for your template, you can select the **Edit Template** option and define them as part of this process.

   If you would like to include a sequence field as part of the record, you must specify this as part of your template Create Attribute values. When you are using a template, information in the **Sequence field position**, **Sequence field length** and **Sequence field increment** fields is ignored.

6. If you have specified the name of a new data set and you want to use another data set as a model, spedify the name of the model data set in the **Like data set** field.

7. Specify your template by doing *one* of the following:
   - In the Copybook or Template **Data set name** and **Member** fields, specify the name of an existing copybook or template, then set the **Copybook or Template** processing option to **1. Above**.
   - Set the **Copybook or Template** processing option to **2. Previous**. This tells File Manager to use the copybook or template that you have most recently associated with this data set (discrete to each user). File Manager ignores the contents of the **Data set name** and **Member** fields.
   - Set the **Copybook or Template** processing option to **4. Create Dynamic**. File Manager ignores the contents of the **Data set name** and **Member** fields and displays the Dynamic Template panel when you press Enter to process the Create Data panel.

8. If you want to run this process as a batch job, select the **Batch execution** option. See “Using File Manager functions in batch jobs” on page 401 for details about using batch mode.

9. Select your **Disposition**, **Use I/O exit**, **Use proc** and **ISPF Packing** Processing Options to suit your needs (see “Creating data without using a template” on page 242 for details).

10. Press Enter.

    If you have specified an existing data set and template, and the process is successful, a message is displayed to show the number of records written to the file.

    If you have elected to create a dynamic template, the Dynamic Template panel is displayed. If you have specified an existing template and selected the Edit Template option, the appropriate template editing panel is displayed. When you have finished defining your template, File Manager either creates the records or asks you to allocate a new data set, depending upon whether you specified an existing or new data set name.

    If you have specified a new data set name, you are asked to allocate it after your template has been specified or created. When the data set has been successfully allocated, the Data Create panel is displayed again, with a message to show the number of records written to the file.

Related topics

“Setting the Data Create Attributes in your template” on page 246
“Specifying a data set and a member name” on page 16
“Allocating a new data set” on page 246

Allocating a new data set

When you specify an output data set that does not exist, File Manager prompts you to allocate the new data set by displaying 2 panels. You select the organization type for the new data set on the first panel. Your choice of data set organization determines which panel is displayed next. You can also provide a "Like data set" to be used as a model to provide data set attributes for the new data set. You can either specify the name of the model data set (which must already exist) directly on the Data Set Create utility panel, or on the new data set Allocation panel. You enter allocation specific information, such as primary and secondary space allocation and record length on the second panel. The second panel allows either the creation of a VSAM data set, a physical sequential or partitioned data set, an HFS file, or an IAM KSDS or ESDS data set.

To allocate a new data set:

1. On the Allocate: New Data Set Organization panel, select the option for the data set organization type.
2. If you want to copy the allocation attributes of an existing data set, specify the name of the data set in the **Existing Data Set** field.
3. Press Enter. The panel that is appropriate for the selected data set organization type is displayed.
4. Complete the fields in this panel to create the data set you require. For general information about defining data sets, see the *DFSMS Using Data Sets* manual appropriate for your operating system. For specific information about valid field values, use File Manager's Field-level Help (put your cursor in the field and press F1).

Related topics

"Creating data without using a template" on page 242
"Creating data sets using a template" on page 244
"Allocate panel" on page 448
DFSMS Using Data Sets

Setting the Data Create Attributes in your template

To enter create attributes for a field:

1. Display your template (dynamic or copybook) in the Field Selection/Edit panel.
   For details on how to do this, see "Editing a template" on page 156.
2. Type `E` in the **Cmd** field adjacent to the relevant field, and press Enter.
   The Field Attributes panel is displayed.
3. Enter values in the Create Attributes fields.
4. Press the Exit function key (F3) to return to the Field Selection/Edit panel.
   An "E" is displayed under the **SHE** column to indicate that the attributes for the field have changed.
5. Save your changes and exit from your template.

Related topics

"Field Attributes panel - alphanumeric fields" on page 553
"Field Attributes panel - numeric field" on page 559

Creating, replacing, and adding data using existing data

When you are in an Edit or View editor session, you can create data elsewhere from the data you are currently editing or viewing.

You can create or replace data in a member of a partitioned data set (PDS or PDSE) or a sequential, VSAM, or z/FS data set, using data from one or more lines of the data in the current editor session.

You can also add data to a sequential, VSAM, or z/FS data set, using data from one or more lines of the data in the current editor session.

- To *create* data in a new member or a new data set using one or more lines of the data in the current editor session, use the CREATE (or CREATEX) primary command. See "Creating a new member or data set using data from an edit session."
- To *replace* data in an existing member or a data set with one or more lines of the data in the current editor session, use the REPLACE (or REPLACEX) primary command. See "Replacing a member or data set using data from an edit session" on page 249.
- To *add* data to an existing data set using one or more lines of the data in the current editor session, use the APPEND (or APPENDX) primary command. See "Adding data to a data set using data from an edit session" on page 250.
- To *create* data in a new member or a new data set using *all* of the data in the current editor session, use the SAVEAS (or SAVEASX) primary command. See "Using all the data from an edit session to create a new member or data set" on page 251.

Creating a new member or data set using data from an edit session

The CREATE primary command (or its abbreviation, CRE) creates a new member of a partitioned data set (PDS or PDSE) or a new sequential, VSAM, or z/FS data set from one or more lines of the data in the current editor session. If the specified member or data set exists, a warning panel is displayed and a choice is presented, allowing you to either replace the existing member (or data set) or to stop processing. (To replace an existing member or data set without being prompted, use the REPLACE primary command. See "Replacing a member or data set using data from an edit session" on page 249.)

To specify the lines to be put into the new member or data set, use one of these:
- C or CC prefix commands (to copy lines)
- M or MM prefix commands (to copy lines, then delete them)
- A label range. (You must specify the "from" and "to" labels on the REPLACE command.)

Note that the lines that are copied or moved to the specified data set or member are only those lines which are *not grouped*. This means that lines in any of these groups are not copied (or moved):
- Excluded groups.
- Suppressed groups (groups of other record types when using a template).
- Not-selected groups (groups of not-selected records when using a template).

This allows for a wide variety of control over what you create in the target member or data set, matching the current SHOW settings in the editor session.

If you specify the member name or data set name as part of the CREATE primary command, and the move or copy line commands (or range operand) are entered, the new member or data set is created immediately from the data in the current editor session. The allocation attributes for a data set created in this way are taken from the data set in the current editor session.

When the current editor session is using a PDS, you can enter the member name with or without parentheses which is taken to mean a member within the same PDS.

You can specify a partially-qualified or fully-qualified data set name as part of the command.

Examples:

CREATE MEMNAM
Creates the new member, MEMNAM.

CREATE (MEMNAM)
Creates the new member, MEMNAM.

CREATE *datasetname***(MEMB1)**
Creates member, MEMB1.

CREATE '*userid.datasetname***(MEMB2)'**
Creates member, MEMB2.

CREATE '*userid.datasetname***'**
Creates data set *userid.datasetname*, based on the current data set attributes.

If you enter the CREATE command without a member name or data set name, or you enter the extended version of this command, CREATEX (or its abbreviation, CREX), File Manager displays an extended create panel allowing you to enter the name of a target data set and, optionally, member. This panel also allows you to decide what data set to use as a model, or, if you want, to manually enter allocation attributes for the target data set. Otherwise, the default data set attributes are taken from the data set on which the editor session is currently operating.

Examples:

CRE .STRT .END
Displays an extended create panel where you can enter the data set name and, optionally, the member name.

CREX '*userid.dataset***'**
Displays an extended create panel where you can enter the data set name and, optionally, the member name.

The **Pack** option on the extended create panel allows you to choose whether the data being created is stored in ISPF packed or standard format. The displayed default reflects the current pack mode of the data being edited or viewed.

After the create processing is completed, File Manager returns you to the existing editor session.

Note:

1. In a CICS environment, CICS resources are not available as a target for this command. Also, MQ resources are not available as a target for this command.
2. If the target file for a CREATE is a KSDS VSAM, the occurrence of duplicate keys causes an informational message to be displayed, but processing of the command continues.

Related topics
"CREATE, CREATEX primary commands" on page 757

"Create panel" on page 512

Replacing a member or data set using data from an edit session

The REPLACE primary command (or its abbreviation, REPL) creates, if it does not already exist, a new member of a partitioned data set (PDS or PDSE) or a new sequential, VSAM, or z/FS data set from one or more lines of the data in the current editor session. If the specified member or data set already exists, it is overwritten. (To display a warning panel that prompts you when you attempt to replace an existing member or data set, use the CREATE primary command. See "Creating a new member or data set using data from an edit session" on page 247.)

To specify the lines to be put into the new member or data set, use one of these:
- C or CC prefix commands (to copy lines)
- M or MM prefix commands (to copy lines, then delete them)
- A label range. (You must specify the "from" and "to" labels on the REPLACE command.)

Note that the lines that are copied or moved to the specified data set or member are only those lines which are *not grouped*. This means that lines in any of these groups are not copied (or moved):
- Excluded groups.
- Suppressed groups (groups of other record types when using a template).
- Not-selected groups (groups of not-selected records when using a template).

This allows for a wide variety of control over what you create in the target member or data set, matching the current SHOW settings in the editor session.

If you specify the member name or data set name as part of the REPLACE primary command, and the move or copy line commands (or range operand) are entered, the new member or data set is created immediately from the data in the current editor session. The allocation attributes for a data set created in this way are taken from the data set in the current editor session.

When the current editor session is using a PDS, you can enter the member name with or without parentheses which is taken to mean a member within the same PDS.

You can specify a partially-qualified or fully-qualified data set name as part of the command.

Examples:

REPLACE MEMNAM
Replaces or creates the new member MEMNAM.

REPLACE (MEMNAM)
Replaces or creates the new member, MEMNAM.

REPLACE *datasetname*(MEMB1)
Replaces or creates member, MEMB1.

REPLACE '*userid.datasetname***(MEMB2)'**
Replaces or creates member, MEMB2.

REPLACE '*userid.datasetname***'**
Replaces or creates data set *userid.datasetname*, based on the current data set attributes.

If you enter the REPLACE command without a member name or data set name, or you enter the extended version of this command, REPLACEX (or its abbreviation, REPLX), File Manager displays an extended replace panel allowing you to enter the name of a target data set and, optionally, member. This panel also allows you to decide what data set to use as a model, or, if you want, to manually enter allocation attributes for the target data set. Otherwise, the default data set attributes are taken from the data set on which the editor session is currently operating.

Examples:

REPL .STRT .END
Displays an extended replace panel where you can enter the data set name and, optionally, the member name.

REPLX '*userid.dataset***'**
Displays an extended replace panel where you can enter the data set name and, optionally, the member name.

The **Pack** option on the extended replace panel allows you to choose whether the data being created is stored in ISPF packed or standard format. The displayed default reflects the current pack mode of the data being edited or viewed.

After the replace processing is completed, File Manager returns you to the existing editor session.

Note:

1. In a CICS environment, CICS resources are not available as a target for this command. Also, MQ resources are not available as a target for this command.
2. If the target file for a replace is a KSDS VSAM, the occurrence of duplicate keys causes an informational message to be displayed, but processing of the command continues.

Related topics
"REPLACE, REPLACEX primary commands" on page 805
"Replace panel" on page 645

Adding data to a data set using data from an edit session

The APPEND primary command (or its abbreviation, APP) adds data from one or more lines of the data in the current editor session to a sequential, VSAM, or z/FS data set.

To specify the lines to be added to a data set, use one of these:
- C or CC prefix commands (to copy lines)
- M or MM prefix commands (to copy lines, then delete them)
- A label range. (You must specify the "from" and "to" labels on the append command.)

Note that the lines that are copied or moved to the specified data set are only those lines which are *not grouped*. This means that lines in any of these groups are not copied (or moved):
- Excluded groups.

- Suppressed groups (groups of other record types when using a template).
- Not-selected groups (groups of not-selected records when using a template).

This allows for a wide variety of control over what you create in the target data set, matching the current SHOW settings in the editor session.

If you specify the data set name as part of the APPEND primary command, and the move or copy line commands (or range operand) are entered, the specified data in the current editor session is appended to the data set immediately.

You can specify a partially-qualified or fully-qualified data set name as part of the command.

Example:

APPEND '*userid.datasetname*'
Appends data to *userid.datasetname*.

If you enter the APPEND command without a data set name, or you enter the extended version of this command, APPENDX (or its abbreviation, APPX), File Manager displays an extended append panel allowing you to enter the name of a target data set.

Examples:

APP .STRT .END
Displays an extended append panel where you can enter the data set name.

APPX '*userid.dataset*'
Displays an extended append panel where you can enter the data set name.

The **Pack** option on the extended append panel allows you to choose whether the data being created is stored in ISPF packed or standard format. The displayed default reflects the current pack mode of the data being edited or viewed.

After the append processing is completed, File Manager returns you to the existing editor session.

Note:

1. In a CICS environment, CICS resources are not available as a target for this command. Also, MQ resources are not available as a target for this command.
2. If the target file for an append is a KSDS VSAM, the occurrence of duplicate keys causes an informational message to be displayed, but processing of the command continues.

Related topics
"REPLACE, REPLACEX primary commands" on page 805
"Replace panel" on page 645

Using all the data from an edit session to create a new member or data set

The SAVEAS primary command creates a new member of a partitioned data set (PDS or PDSE) or a new sequential, VSAM, or z/FS data set from *all* the lines of the data in the current editor session. If the specified member or data set exists, a

warning panel is displayed and a choice is presented, allowing you to either replace the existing member (or data set) or to stop processing.

Note that the lines that are copied to the specified data set or member are only those lines which are *not grouped*. This means that lines in any of these groups are not copied:
- Excluded groups.
- Suppressed groups (groups of other record types when using a template).
- Not-selected groups (groups of not-selected records when using a template).

This allows for a wide variety of control over what you create in the target member or data set, matching the current SHOW settings in the editor session.

If you specify the member name or data set name as part of the SAVEAS primary command, the new member or data set is created immediately from the data in the current editor session. The allocation attributes for a data set created in this way are taken from the data set in the current editor session.

When the current editor session is using a PDS, you can enter the member name with or without parentheses which is taken to mean a member within the same PDS.

You can specify a partially-qualified or fully-qualified data set name as part of the command.

Examples:

SAVEAS MEMNAM
Replaces or creates the member, MEMNAM.

SAVEAS (MEMNAM)
Replaces or creates the member, MEMNAM.

SAVEAS *datasetname*(MEMB1)
Replaces or creates member, MEMB1.

SAVEAS '*userid.datasetname***(MEMB2)'**
Replaces or creates member, MEMB2.

SAVEAS '*userid.datasetname***'**
Replaces or creates data set *userid.datasetname*, based on the current data set attributes.

If you enter the SAVEAS command without a member name or data set name, or you enter the extended version of this command, SAVEASX, File Manager displays an extended saveas panel allowing you to enter the name of a target data set and, optionally, member. This panel also allows you to decide what data set to use as a model, or, if you want, to manually enter allocation attributes for the target data set. Otherwise, the default data set attributes are taken from the data set on which the editor session is currently operating.

Examples:

SAVEAS Displays an extended Saveas panel where you can enter the data set name and, optionally, the member name.

SAVEASX '*userid.dataset***'**
Displays an extended Saveas panel where you can enter the data set name and, optionally, the member name.

The **Pack** option on the extended Saveas panel allows you to choose whether the data being created is stored in ISPF packed or standard format. The displayed default reflects the current pack mode of the data being edited or viewed.

After the saveas processing is completed, the editor session you are returned to is an edit of the target PDS member or data set as specified, and the previous member or data set is released without any save occurring. Also note that any starting position, record limits, or sampling information that may have been previously specified, is not active for this edit of the SAVEAS target data set.

Note:

1. In a CICS environment, CICS resources are not available as a target for this command. Also, MQ resources are not available as a target for this command.
2. If the target file for a SAVEAS is a KSDS VSAM, the occurrence of duplicate keys causes an informational message to be displayed, but processing of the command continues.

Related topics
"SAVEAS, SAVEASX primary commands (data)" on page 813
"Saveas panel" on page 646

Copying data sets

When you are using File Manager under ISPF, you can use the Copy Utility (option 3.3) to copy data from one data set to another data set; when you are programming a batch job, REXX procedure or TSO clist, you can use the equivalent function, DSC (Data Set Copy). Whether you are using the panel or the function, the Copy Utility (option 3.3) lets you copy data using no templates, a "From" template only, or both "From" and "To" templates, in order of increasing flexibility.

You can also invoke the Copy Utility by issuing the COPY line command (or by selecting **Copy** from the Process pull-down menu) on the Data Set List panel, reached using the Catalog Services Utility (3.4). For more information, see "Working with a list of catalog entries" on page 314.

With or without templates, you can enhance File Manager copy processing by specifying a REXX procedure or DFSORT statements. Using a procedure potentially allows you greater flexibility than any of the choices listed here, but it involves the extra complexity of programming.

With or without templates, you can copy records where the input and output data sets have different:
- Record formats
- Record lengths
- Block sizes

You can use the Copy Utility to:

- Copy data from any supported data set to any other supported data set. Concatenated like and unlike sequential data sets are supported. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for copy processing. Such invocation may fail since DFSORT does not allow for unlike concatenation of data sets. In such cases, the DFSORT use may be disabled with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

Concatenated partitioned data sets with like attributes are fully supported. Mixed partitioned data sets with like and unlike attributes are processed correctly providing that the members being selected for processing come from libraries with matching attributes, otherwise errors may be reported.

- Select the records to be copied using the start key (VSAM only), skip and copy count fields, or a conditional expression defined in an input template.
- Change file attributes. You can copy records where the input and output data sets have differing record formats, record lengths or block sizes. The copy process truncates or pads records appropriately. To specify a pad character to be used, use the **PAD** processing option on the Set Processing Options (option 0) panel.
- Copy selected fields, change the size of fields, and create new fields in the output file by using a "From" template with a "To" template.
- Generate data in external format
- Allocate a non-VSAM data set, or define a VSAM data set.
- Copy sequential data sets or PDS or PDSE members, converting the data to or from ISPF PACK data format.

Note: File Manager supports the copying of Load Modules, when the following conditions are met:

- Your input and output data sets are PDS or PDSEs.
- Your TSO environment is active (and you can use the TSO authorized program services), or you are running File Manager as program-authorized.
- You have not specified a REXX or DFSORT user procedure.
- You have not specified Start key, Skip or Copy counts.
- You are not using templates.
- You do not request member record counts.

The tasks described in this section are:

"Copying without using a template"
"Selecting a range of PDS(E) members" on page 34
"Renaming members as you copy" on page 258
"Copying using a copybook or template" on page 259
"Copying records with segmented data templates" on page 262
"Copying files or members containing JCL" on page 267
"Copying records with segmented data templates" on page 262
"Generating data in external format - XML representation" on page 263
"Using a DFSORT or REXX procedure" on page 267
"REXX member selection" on page 268

Copying without using a template

When you are copying data sets and do not use a template, all data in an input data set record is copied to the output data set. You can limit the number of records selected by using the **Start key**, **Skip count** and **Copy count** fields on the Copy Utility (option 3.3) panel (for example, you can specify the number of records to be copied).

If you are *not* using a template for your copy, use the Copy Utility panel (see Figure 142 on page 492) as follows:

1. From the Primary Option Menu panel, select option **3. Utilities** and then option **3. Copy**.

   The Copy Utility panel is displayed.

2. Perform either of these actions
   - Specify the data set that contains the data you want to copy, using a combination of the **Data set name**, **Member**, and **Volume serial** entry fields as appropriate.
   - To select a range of members based on selection criteria, specify the data set name or name pattern for a PDS(E) data set, optionally the member name or pattern, and select the **Advanced member selection** option

     Note: If you enter details in the **Member** field and select the **Advanced member selection** option, File Manager populates the **Member name** field on the subsequent Advanced Member Selection panel with the same value.
3. Set your record-level selection, using the following entry fields on the Copy Utility panel:

   Start key
   If required, specify a **Start key** for the data (VSAM only). Copying starts at the first record with a key or slot number greater than or equal to the specified value. If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. Keys may also be in hexadecimal format (e.g. X'0102').

   Skip count
   If required, specify a **Skip Count** for the data. Copying starts after skipping the number of records specified in the field.

   Include
   If required, specify the number of physical records that to be included in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

   Repeat skip
   If required, specify the number of physical records to be skipped in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

   Copy count
   Specify the number of records to be copied. Enter ALL to copy all records.

   Note: The Start key and Skip count fields are mutually exclusive.
4. Select **2. None** in the **Copybook/template usage** field.

   Note: When this field is set to None, data in the **Copybook or Template Data set name** and **Member** fields is ignored.
5. Select the processing options that you want:

   Batch execution
   Select this option when you want to generate the JCL needed to run the function in batch. The generated JCL is displayed in an Edit session.

   Use proc
   Select this option when you want to run a REXX procedure or DFSORT statements in conjunction with your copy action, and then specify the procedure name in the adjacent field.

   To create a temporary procedure for one-time use enter an asterisk (*). File Manager displays an Edit panel that you can use to create a new procedure.

To use an existing procedure, specify the name of the member containing the procedure you want to use. The member must belong to the PDS allocated to ddname FMNEXEC. You can enter the name of the member or a member name pattern (other than *). If you specify a pattern, the Member Selection panel for the PDS is displayed when you press Enter. You can then select the required member by entering an S in the **Sel** field.

Ignore length mismatch
Select this option when you want to include records in the copy process that are shorter than the matching template structure length.

Use I/O exit
Select this option when you want to run a user I/O exit that processes the "From" (input) data set in conjunction with the copy action. If selected, specify the exit name in the adjacent field.

Note: This option is only available when File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field in the Set System Processing Options panel is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.

Do not select this option if you plan to use the ISPF Packing option on the next panel.

Skip member selection list
Select this option when you have either used a generic name in the Member field or specified Advanced member selection, and want to copy all members that match the name or range of names, without further refining the member list. This option is ignored if errors are found whilst copying.

Directory integrity
Forces an override of the default PDS(E) member processing method which allows for faster PDS directory access.

This option has significant performance impact. When selected, the members are processed in a way which allows concurrent directory updates as File Manager accesses the members using current directory information.

When not selected, the member processing is performed faster, but may be affected by PDS(E) directory updates, possibly causing I/O errors if the data set is updated concurrently.

Report PDS record counts
Select this option when you want the count of records for copied PDS(E) members to be displayed (online) or printed (batch).

6. Press Enter. The Copy To panel is displayed.
7. Type the "To" data set details. Use a combination of the **Data set name**, **Member**, and **Volume serial** entry fields (the volume serial is only required for data sets which are not accessible via the system catalog) to specify the data set (and PDS(E) member if appropriate) to which you want to copy the data.

   If the output data set does not exist, File Manager displays the Allocation panels to help you create a new data set.
8. Select the processing options for the Copy To panel:

DISP Determines the disposition of the "To" (output) data set. Specify OLD or MOD.

1. OLD
Select this option when you want the copied records to be placed into the "To" data set, starting from the beginning of the data set and overwriting any existing records.

2. MOD
Select this option when you want the copied records to be appended to the end of an existing output data set.

Note: You cannot append records to a PDS or PDSE member or to a packed sequential data set. You also cannot append packed records to a non-packed sequential data set. You must use the OLD option to replace these records.

Note: SMS might modify the allocation of new data sets on your system. For details, contact your SMS Administrator.

Replace members
Select this option when you want to replace like-named members in an output partitioned data set.

Use I/O exit
Select this option when you want to run a user I/O exit that processes the "To" (output) data set in conjunction with the copy action. If selected, specify the exit name in the adjacent field.

If you specify an I/O exit, you cannot use the ISPF Packing option.

ISPF Packing
Provided that the output data set is a sequential, PDS or PDSE file, an I/O exit routine is not used and the DISP is set to OLD, select one of these options to control the copy behavior when processing data that is in ISPF PACK format.

1. Asis
Select this option when you want File Manager to write the output in the same format (packed or unpacked) as the input records. If the input data is packed, it is unpacked for the processing operation, and then written out in ISPF PACK format.

2. Unpack
Select this option when you want File Manager to write the output in unpacked format, regardless of the format of the input records. If the input data is packed, it is unpacked for the processing operation, and then written out without the ISPF PACK format.

3. Pack
Select this option when you want File Manager to write the output in ISPF PACK format, regardless of the format of the input records. If the input data is packed, it is unpacked for the processing operation, and then written out in the ISPF PACK format.

4. None
Select this option when you do not want File Manager to check

for packed records or unpack records for processing. If the input data is packed, any processing actions operate on the records in their packed format.

This is the only option allowed when an I/O exit has been specified, when the Disposition is set to MOD or when the Copy To data set is not a sequential file or PDS(E) member.

5. **Skip**

   Select this option when you want to halt the copying or processing actions, if the input data is packed.

9. Press Enter.

   If you selected the **Advanced member selection** option, File Manager displays the Advanced Member Selection panel:

   a. Specify the selection criteria to select the members you want to copy and press Enter.
   b. If you have left the **Member name** field on the Advanced Member Selection panel blank or have entered an asterisk (*) or a mask, (and you did not select the **Skip member name list** on the Copy From panel), File Manager displays the Member selection panel. In this case, select the members you want to copy (either by typing an "S" in the **Cmd** field, or with the SELECT primary command) and press Enter.

   File Manager copies the selected data in the "From" data set to the "To" data set.

Care must be taken when attempting to combine the **Use proc**, **Use I/O exit** and **ISPF Packing** options. You cannot use an I/O exit and the ISPF Packing option at the same time. You can provide a procedure and use an I/O exit or the ISPF Packing option but you cannot include *Fastproc DFSORT statements in your procedure.

Related topics

"Allocating a new data set" on page 246
"Copy From panel" on page 491
"Copy To panel" on page 497
"Specifying a data set and a member name" on page 16
"DSC (Data Set Copy)" on page 875
"Selecting a range of PDS(E) members" on page 34
"Advanced Member Selection panel" on page 439

Renaming members as you copy

You can use the Copy Utility to change the names of multiple members in a PDS(E), as you copy members from one data set to another. For example, you might want to copy a range of members named TEST*xxx* to a new data set and rename all of these as PROD*xxx*.

To rename members as you copy:

1. In the Copy From panel, enter a member name mask in the **Member** field.

   OR

   Select the **Advanced member selection** option and specify a range of member names.
2. Specify the remainder of your copy options, as required, then press Enter.
3. In the Copy To panel, enter a new member name mask in the form of a member name pattern.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

4. Press Enter. The new member names are displayed in the Prompt field on the Member selection panel.

```
Sel Name     Prompt   Alias-of Size Created
_   DATA1    TEST1             43 01/01/09
_   DATA2    TEST2             21 02/09/06
_   DATA3    TEST3             40 02/09/06
_
```

5. Select the members that you want to copy and press Enter. The Prompt field entry changes to `*COPIED` for the members that were successfully copied and renamed.

Copying using a copybook or template

When you choose to use templates in your copy action, you can specify a "From" template only or both "From" and "To" templates.

"From" template only
Excludes not-selected records from the output data set.

You can use the "From" template to exclude not-selected input records from being copied to the output data set. During the copy, the Copy Utility uses the "From" template to provide information about the selected record types, record identification criteria, record selection criteria and record structure. Once selected, the entire contents of the record are copied. Any field selection and field resequencing settings in the "From" template are ignored for copying, as field mapping and field create attributes are taken from the "To" template.

You determine which records are selected or excluded by editing the template and:
- selecting record types;
- specifying record identification criteria; and/or
- specifying record selection criteria.

Not-selected records:
- do not match a record type in the template, due to either record identification criteria or record length;

- match a record type, but belong to an unselected record type; or
- do not meet the record selection criteria for the record type.

"From" and "To" templates (based on the same copybook or dynamic template structure)

Allows you to specify how fields in the input data set are mapped to the output data set: to the same field (this is the default), to a different field or unmapped.

You can use the “To” template to specify field mapping to determine which fields in the input data set are copied to the fields in the output data set. Unmapped fields are initialized in the output data set according to “create attributes” in the “To” template. (If no create attributes have been specified, then unmapped numeric fields are initialized to zero and unmapped alphanumeric fields are initialized to blanks.)

File Manager ignores any record type selection, record identification criteria or record selection criteria in the “To” template, as this information is drawn from the “From” template.

The only information in the “To” template that is used for copying is the record structure, field mapping and field create attributes.

"From" and "To" templates (based on different copybook or dynamic template structures)

Allows the input and output data sets to have different record structures, according to the differences in the copybook or dynamic template.

The Copy Utility inserts, deletes or moves fields, or changes field lengths or data types, according to the differences between the templates. For example:

- To insert a field, use a “To” template that is identical to the “From” template, except for an inserted field definition.
- Similarly, to delete a field, use a “To” template that is identical to the “From” template, but with the unwanted field removed.

Typically, you create the “To” template by copying the “From” copybook or dynamic template, then you edit the “To” copybook (or dynamic template) so that it describes the record structure that you want in the output data set.

The table below summarizes how the Copy Utility uses information in the “From” and “To” templates:

The Copy Utility determines...	Using this information...	In this template...	
		From	To
Which records are copied	Selected record types Record identification criteria Record selection criteria	✓	
Which fields are copied, and to where	Field mapping		✓
How new (or unmapped) fields are initialized	Create attributes		✓

By default, the Copy Utility maps fields in the “From” template to fields with the same name in the “To” template. You can edit the field mapping in the “To” template, and either delete the mapping for a field (so that its value in the output

data set is initialized, rather than copied from the input data set), or specify yourself which field in the "From" template is mapped to a field in the "To" template.

The Copy Utility truncates or pads records appropriately. You can specify the pad character in the **PAD** field on the Set Processing Options (option 0) panel.

If you *are* using a copybook or template for your copy, use the Copy Utility panel (see Figure 142 on page 492) as follows:

1. From the Primary Option Menu panel, select option **3. Utilities** and then option **3. Copy**.

   The Copy Utility panel is displayed.
2. Enter the "From" data set details.
3. Enter the "From" copybook or template details. Use a combination of the **Data set name** and **Member** entry fields to specify the copybook or template that describes the data in the "From" data set.

   You can select data for copying at either record level or field level:

   - For record-level selection, set the record identification and record selection criteria in your "From" template.
   - For field-level selection, use a "From" template with a "To" template, and specify the selected fields, field attributes and field mapping in your "To" template.

   If you specify both record-level and field-level selection, File Manager first selects data at record level, then at field level.
4. Select Copybook/template Processing Option **1. Above** or **3. Create dynamic** press Enter.

   If you selected option 1, an extended version of the Copy To panel appears. This form of the panel allows you to specify the "To" copybook or template.

   If you selected option 3, then you must create the dynamic template. Once you have done so, the extended Copy To panel is displayed.
5. Type the "To" data set details.
6. Select Copybook/template Processing Option **1. Above**, **2. None**or **3. Create dynamic** press Enter.
7. If you selected option 1, enter the "To" copybook or template details. Use a combination of the **Data set name** and **Member** entry fields to specify the copybook or template that contains field selection and mapping information for your "To" data set.
8. If required, select the **Edit template mapping** option. When the template is displayed, you can specify the mapping of input fields to output fields the data creation patterns for new fields.
9. Press Enter. File Manager copies the selected data in the "From" data set to the "To" data set.

Related topics

"Copying without using a template" on page 254
"Specifying a data set and a member name" on page 16
"Mapping fields between templates" on page 184
"DSC (Data Set Copy)" on page 875

Scrambling data

Scrambling data allows you to create test data based on production (or "live") data, but with the ability to change the values of certain fields. In this way, you can avoid sensitive or confidential information appearing in test data.

When can you scramble data?

When you use the Copy Utility (option 3.3) to copy data from one data set to another data set, you can choose to scramble some or all of the fields being copied.

For scrambling to occur during the copy process, you must supply an *output template* which has mapped fields *marked for scrambling*.

Scrambling rules

File Manager scrambles data according to these rules:

- If you do not specify any value or range options:
  - Uppercase alphabetic characters are scrambled to other uppercase alphabetic characters.
  - Lowercase alphabetic characters are scrambled to other lowercase alphabetic characters.
  - DBCS characters are scrambled to other DBCS characters.
  - Numerics are scrambled to other numerics.
  - Any other characters remain unchanged.
  - Repeatable scrambling produces unique results for numeric fields defined with **Leading zeros** set to YES, and all non-numeric and non-DBCS fields.
- If you provide a value list, then the field is populated with a value from the list.
- If you provide a range, then the resultant number is in the range provided.
- If you specify a scramble type of random or repeatable and also the value option and a value data set, then the field is populated with a value from the data set.
- If you specify a scramble type of translate, then the input field value is matched and the corresponding output value is used from the value data set.

Related topics

"Specifying scrambling options" on page 180
"Field Attributes panel - alphanumeric fields" on page 553
"Field Attributes panel - numeric field" on page 559
"Value List Edit panel" on page 705
"Scramble Exit Specification panel" on page 646

Copying records with segmented data templates

Previous releases of File Manager that copied data with a segmented template, desegmented the output file. This behaviour has been changed so segmented data can be copied, and the record segment relationships, or physical record structures, are maintained. This means you can:

- Selectively copy records from a segmented data set using an input template with selection criteria.
- Use an input and output template to reformat segmented data. This includes the ability to scramble output. Note that the output segment lengths will be determined by the output template, and input segments will be padded or truncated during reformatting. To ensure you do not get unexpected results, make sure you do not have length errors when mapping input segments.
- Use a procedure to drop or insert segments within a physical record. Procedure processing sees the input data one segment at a time. Coding a RETURN 'DROP' will remove that segment from the current output record. Coding a WRITE()

statement inserts a segment on the current output record. Coding a RETURN 'STOP' will only write out the current physical record if the segment being processed is the last segment on that physical record, otherwise it behaves like a RETURN 'STOP IMMEDIATE' where the current physical record being processed is not written. See given examples:

Example 1.

Insert a new segment in the output record.

Consider the following input segmented record:

```
AsegBsegCsegDseg
```

is mapped by a segmented template with 5 layouts mapping segments A to E:

```
$$FILEM DSC ,
$$FILEM TCIN=MYSEG.TEMPLATE(SEGTP),,
$$FILEM PROC=*
IF FLD(1,1) = 'D' THEN DO              /* found a Dseg */
  WRITE()                              /* Write out current segment */
  OVLY_OUT('E',1)                      /* Create 'Eseg' in output area */
END                                    /* normal return writes Eseg  */
```

After running the above procedure, the output record would be

```
AsegBsegCsegDsegEseg.
```

Example 2.

Delete a segment in the output record.

Consider the following input segmented record:

```
AsegBsegCsegDseg
```

```
$$FILEM DSC ,
$$FILEM TCIN=MYSEG.TEMPLATE(SEGTP),,
$$FILEM PROC=*
IF FLD(1,1) = 'C' THEN                  /* found a Cseg */
  Return 'DROP'                         /* drop this segment */
```

After running the above procedure the output record would be

```
AsegBsegDseg
```

Example 3.

Desegment a segmented file.

```
//DDIN     DD DISP=SHR,DSN=MYSEG.DATA
//DDOUT    DD DUMMY
//DDDESEG  DD DISP=SHR,DSN=MYDESEG.DATA
//SYSIN    DD *
$$FILEM DSC ,
$$FILEM TCIN=MYSEG.TEMPLATE(SEGTP),
$$FILEM PROC=*
WRITE(DDDESEG)   /* Write segment out as physical record */
RETURN 'DROP'    /* don't copy any data */
/*
```

Related topics

"Segmented data templates" on page 131
"Setting up a template to be used with segmented data" on page 151

Generating data in external format - XML representation

A traditionally formatted data set or file, with format described by a template (copybook), can be represented in an external format. This is possible by applying external format rules to the data in a character format determined by the input

template. The output of a copy operation has an external format which can be recognized by applications on various platforms.

Currently, you can request that the output of the copy operation be "well- formed" XML data (the concept and terminology of Extensible Markup Language can be found in the XML Specification 1.0 from the World Wide Web Consortium, www.w3.org). The output format derived from the input template also conforms to XML rules. A typical record in the output file is an XML line corresponding to an elementary data item in the input record, and has the format:

```
start-tag content end-tag
```

where *start-tag* and *end-tag* include the data item name and *content* is a character representation of the data item.

Using the copybook shown in “About copybook templates” on page 129:

```
01 ORDERS.
   05 ORDER-ID        PIC X(5).
   05 CUSTOMER-ID     PIC X(5).
   05 ORDER-DATE.
      10 ORDER-YEAR   PIC 9(4).
      10 ORDER-MONTH  PIC 9(2).
      10 ORDER-DAY    PIC 9(2).
```

and the input record with an Order ID of O1002 and a Customer ID of C0015 made on 20050110, the corresponding sequence of XML lines is:

```
<ORDER>
  <ORDER-ID>O1002</ORDER-ID>
  <CUSTOMER-ID>C0015</CUSTOMER-ID>
  <ORDER-DATE>
    <ORDER-YEAR>2005</ORDER-YEAR>
    <ORDER-MONTH>01</ORDER-MONTH>
    <ORDER-DAY>10</ORDER-DAY>
  </ORDER-DATE>
</ORDER>
```

File Manager assumes that the names used in a copybook or template are legitimate XML names according to the XML specification. It does not check that each data name, other than FILLER, is unique within the containing group. However, names which do not start with a letter or an underscore are prefixed with an underscore.

The format of elementary data depends on the input template or copybook. The result is similar to the output from the Print Utility. Numeric data is converted to character representation. File Manager removes leading zeros and blanks, trailing blanks, skips a positive sign, and so on. Character data is included into the XML content with trailing blanks removed (leading blanks are significant in data and are not suppressed).

There are situations when data cannot be represented in the XML output by its value. Data may not match the data type requirements. For example, when a numeric field contains non-numeric characters. Character data can contain non-printable and special characters. Non-printable characters are defined by the default (or customized) FMNTRTBS translation table. The following special characters have special meaning in the XML output and require different representation:

">" (greater than)
"<" (less than)
"'" (apostrophe)

""" (quote)
"&" (ampersand)

The XML output is generated in the character set of the input data (EBCDIC, DBCS), but can optionally be converted to Unicode. File Manager uses standard z/OS support for Unicode, assuming that appropriate infrastructure and services are present. Conversion to Unicode is possible if the conversion environment is created and activated (for details, see the *z/OS Support for Unicode Using Conversion Services*). File Manager assumes CCSID 0037 for EBCDIC, 0939 for DBCS (MBCS), and 1200 for Unicode.

You can tailor the format and content of the XML output by specifying:

- How to represent special characters
- How to represent non-printable characters
- How to represent invalid data
- Whether FILLER should be included or ignored
- Whether redefinitions are to be included or ignored
- Whether to get the XML output in the character set of input data (EBCDIC, DBCS) or in Unicode
- The number of blanks used to indent each logical level of XML tag nesting for better readability (logical levels 1, 2, 3, and so on), are assigned by FM by renumbering user levels specified in the copybook (for example 01, 05, 10, and so on)

The XML output contains additional information included as separate lines, or attributes following data-element names. They document:

- The input data set name and the template (copybook). For example:

  ```
  <INPUT FILE="FMN.SEQ1" FORMAT="FMN.TEMPLATE(SEQ1)">
  ```

- The record sequence number and, in the case of keyed data, the key value. For example:

  ```
  <ORDER SEQ_NUMBER="998" KEY="01002">
  ```

- The occurrence number, if the data element is an array. For example:

  ```
  <MONTH-END ITEM_NUMBER="(2)">24</MONTH-END>
  ```

- Non-printable or special characters found in the content of the element and converted according to processing parameters. For example:

  ```
  <EMP-ID NONPRINT_CHAR="REPLACE">AA..17</EMP-ID>
  <JOB-ID SPECIAL_CHAR="ESCAPE">A15&B32</JOB-ID>
  ```

- Any invalid data conditions met when converting a numeric data to a character string; the element is presented according to processing parameters. For example, skipped:

  ```
  <EMP-SALARY INVALID_DATA="SKIP"></EMP-SALARY>
  ```

In the cases of invalid data, non-printable and special characters, the attributes document the actual processing of particular data rather than the parameters specified. For example, if a data field contains both non-printable and special characters, with SKIP and REPLACE as the corresponding parameters, the content is skipped. REPLACE is inconsistent with SKIP in this case and is ignored. In general, if a data field contains both special and non-printable characters, SKIP and HEX have precedence over all other options. CDATA has less priority than SKIP and HEX, but higher priority than the rest of the options.

The XML output can be any data set allowed by the Copy Utility, except for VSAM KSDS. The output data set may be fixed or variable in length. If you want an output record created for each XML line, make sure that the maximum logical record length (LRECL) can contain the largest generated XML line. The size of the line depends on the names used in the template, the maximum length of the character representation of values, and the way the special and non-printable characters and invalid data are processed. Some special characters are substituted with strings, some data presented in hexadecimal form, and so on. This can make the line considerably longer than expected. If you want each XML line to be placed in a single output record, and the record is not long enough for a particular XML line, File Manager truncates the XML line, ends processing and reports an error (and, if in batch, sets a tailorable condition code of 8 indicating a truncation error). If you decide that an XML line can span multiple output records, the output dataset can have any logical record length.

To generate XML-formatted output, use the Copy From panel as follows:

1. From the Primary Option Menu panel, select option 3 (Utilities), and then option 3 (Copy). File Manager displays the Copy Utility panel.
2. Enter the "From" data set details.
3. Enter the "From" copybook or template details. Use a combination of the **Data set name** and **Member** entry fields to specify the copybook or template that describes the data in the "From" data set. You can select data for copying at either record level or field level:
   - For record-level selection, set the record identification and record selection criteria in your "From" template.
   - For field-level selection, use a "From" template with a "To" template, and specify the selected fields, field attributes and field mapping in your "To" template.

   If you specify both record-level and field-level selection, File Manager first selects data at record level, then at field level.
4. Select **Export mode** to indicate that you want output in an external format.
5. Select Copybook/template Processing Option 1 (Above) or 3 (Create dynamic) press Enter.

   If you selected option 1, File Manager displays an extended version of the Copy To panel. This form of the panel allows you to specify the additional options.

   If you selected option 3, then you must create the dynamic template. Once you have done so, File Manager displays the extended Copy To panel.
6. Type the "To" data set details.
7. Optionally, customize the generation of output. For XML, you can affect the generation and readability by specifying how to represent non-printable characters and invalid data, whether to include fillers and redefines, how to indent when nesting successive levels of XML tag nesting, and so on.
8. Press Enter. File Manager generates the selected data from the "From" data set and writes it, in XML format, to the "To" data set.

Related topics

"Specifying a data set and a member name" on page 16
"Copy From panel" on page 491
"Copy To panel" on page 497
"DSC (Data Set Copy)" on page 875

Copying files or members containing JCL

If the file or PDS(E) members you are copying contain JCL, you can instruct File Manager to maintain the integrity of the JCL syntax where the contents of the file or PDS(E) member have been changed during the copy process.

To maintain the integrity of JCL syntax as you copy, in the Copy From panel:

1. Select the **JCL Source format** option.
2. Select the **Use proc** option.
3. Specify the remainder of your copy details as required and press Enter.
4. Specify details on the Copy To panel as necessary and press Enter.

   File Manager copies the selected data in the "From" data set to the "To" data set, maintaining the integrity of any JCL where the JCL is changed during the copy process.

Related topics

"Copy From panel" on page 491
"Copy To panel" on page 497
"DSC (Data Set Copy)" on page 875
"Working with files or members containing JCL" on page 279

Using a DFSORT or REXX procedure

You can specify an existing DFSORT or REXX procedure, or create a new one, to further enhance the way in which data is selected and copied.

You can code any number of DFSORT or REXX statements and functions to manipulate the output record, select specific records, print reports and tally numeric values.

When you are coding a REXX procedure, be aware of the following:

- A copy procedure is run after record-level selection, and any field selection supplied in a template, has been performed.
- If the record has been reformatted as a result of the template processing, the variable INREC contains the input record value and the variable OUTREC contains the reformatted output record.
- If a Copy count is specified, this can affect the number of records presented to the REXX procedure. The Copy count only applies to the number of records written to the primary output data set. It does not apply to records written in the REXX procedure with the WRITE() function.
- When using the PRINT() function with field formatting (TABL or SNGL), the template is determined as follows:
  - If an input template without an output template has been specified, the input template is used.
  - If an output template is specified, and the record value specified to print is the input record, the input template is used. Otherwise the output template is used.
  - If a record cannot be matched against a record layout in the selected printing template, it is not printed.

  Ensure the record value matches the template that File Manager uses to print the data set.

Related topics

Chapter 13, “Enhancing File Manager processing,” on page 405

Example 1

This shows a DFSORT procedure that copies records, inserting 40 bytes of blanks, following by 40 bytes of the input file, beginning from position 41.

```
*FASTPROC
 OUTREC FIELDS=(1,40,40X,41,40)
```

Example 2

This shows a REXX procedure that copies records and writes type 01 to DD01, type 02 to DD02, type 03 to DD03, and the rest to the default output file. The type starts in column 10 for a length of 2.

```
  ddname = 'DD'||fld(10,2)        /* DDname for WRITE function */
  if nco(fld(10,2),1,2,3) then do /* does it contain 1,2,3     */
      write(ddname)               /* Yes Write to DDnn         */
      return 'DROP'               /* don't copy to output DD    */
  end
```

Note: The records that are not processed by the procedure are, by default, copied to the primary output data set.

Example 3

This shows a combination of DFSORT and REXX statements that copies records with type A and prints the first 10. The type value is in column 6.

```
*FASTPROC
 OMIT COND=(1,6,CH,NE,C'A')
*REXXPROC
 If prtcount() <= 10 then print(inrec,'CHAR') /* Print 10 */
    RETURN                                    /* Copy     */
 return 'DROP'                                /* Drop the rest */
```

Example 4

This shows a REXX statement that changes the output records to uppercase while copying.

```
  upper outrec
```

REXX member selection

For PDS or PDSE data sets, you can determine whether a member should be copied or not copied, depending upon a condition being true within the member records. This conditional test is supplied to File Manager via a REXX procedure that includes at least one of the following RETURN strings:

RETURN PROCESS MEMBER
: Indicates that the entire member is to be included in the copy. No more records in this member are passed to the REXX proc. The member is copied intact, subject to any specified template processing, which is performed before the user REXX proc is invoked.

RETURN DROP MEMBER
: Indicates that the member is to be excluded from the copy. No more records in this member are processed. Processing continues with the next member.

When REXX member selection is in effect, records are read from an input member and then cached in memory until the decision is made whether the member is to be copied. Once the decision has been made, the entire member is either copied or dropped, depending upon the return string specified.

If the entire member is processed without encountering a DROP MEMBER or PROCESS MEMBER return string, the member is processed according to the specified default action.

If either of these strings is returned by the REXX processing when REXX member selection HAS NOT been specified, it is treated as if it was a RETURN with no argument strings, except that a warning message is issued. Subsequent records continue to be passed to the REXX proc. Similarly, if a DROP, STOP or STOP IMMEDIATE string is returned by the REXX processing when REXX member selection HAS been specified, it is treated as if it was a RETURN with no argument strings, except that a warning message is issued. Subsequent records continue to be passed to the REXX proc until a decision has been made on whether to DROP or PROCESS the member.

Other REXX statements that control the dropping or updating of records are not performed when REXX member selection is used. Any changes to the special File Manager-defined REXX variable OUTREC are ignored. However, REXX statements that do not involve updating or dropping records are performed. This may lead to some unexpected results. In general, if you need to copy members of a PDS conditionally based on the contents of the member, and also want to manipulate the contents of the member, try to execute this task as a two-step process using separate REXX procedures.

To specify conditional member processing:

1. On the Copy Utility - Copy From panel, specify your data set name and other copy options.
2. Select the **REXX member selection** option and specify the default action (P or D) in the adjacent field. If this field is left blank, P is assumed. P denotes process or copy the member, D specifies drop or do not copy the member.
3. Select the **Use proc** option and specify an existing REXX procedure or enter an * in the field to create a new REXX procedure. This procedure must contain a conditional test and at least one of the relevant RETURN strings.
4. Complete the copying process as required.

Example 1

In this example, the SUBSTR function is used to test records in each member being processed. When the procedure encounters a record in which the condition tests as True, the entire member is dropped from processing. All other members are copied or not copied, according to the default setting on the Copy From panel.

```
if substr(inrec,4,3) == '333'
then RETURN DROP MEMBER
return
```

Example 2

This example demonstrates what happens when you use REXX member selection at the same time as other REXX procedure statements. In this scenario, you are attempting to copy members in a data set and to tally a value in those members at the same time.

The first member (MEM1) contains 5 records:

```
AAA111BBB456CCC789
AAB222BBB456CCC789
AAC333BBB456CCC789
AAA444BBB456CCC789
AAA555BBB456CCC789
```

The second member (MEM2) contains 4 records:

```
AAA001BBB456CCC789
AAA002BBB456CCC789
AAA003BBB456CCC789
AAA004BBB456CCC789
```

The Copy Utility **REXX member selection** default is set to 'P', and the following procedure is supplied.

```
/* rexx */
if substr(inrec,1,3) == 'AAA'  /* If cols 1-3 in current rec contain 'AAA' */
then x = tally(4,3,Z,'total') /* then add up the value in columns 4-6 */
if substr(inrec,4,3) == '333'  /* If cols 4-6 contain '333' */
then RETURN DROP MEMBER       /* then drop member from copy processing */
if substr(inrec,4,3) == '003'  /* If cols 4-6 contain '003' */
then RETURN PROCESS MEMBER    /* then process (copy) member */
return
```

The tally for "total" includes the first record from MEM1, even though MEM1 is dropped from the copy process. The second and third records don't satisfy the 'AAA' requirement and so aren't tallied, and the subsequent records won't be processed because the decision to drop has been made while processing the third record.)

The tally for "total" also includes the first three records in MEM2. The fourth record is not processed by the REXX proc because the decision to copy has been made while processing the third record.

The end result of this copy action is that MEM1 is dropped, MEM2 is copied, and the tally for "total" equals 117. (111 + 1 + 2 + 3).

Related topics

"Copy From panel" on page 491
"RETURN return values" on page 1161
"External REXX functions" on page 1094

Finding and changing data in multiple PDS members

When you need to search for or change a string in more than one PDS member, the Find/Change Utility provides an efficient way to work with multiple members. You can use the panel to select and refine a list of data set members, then enter FIND or CHANGE commands on the Command line. While you can use this utility to find and change data in VSAM and sequential data sets, it might be simpler to make changes in these types of data sets via the Edit panel.

You can also use the Find/Change Utility in conjunction with a REXX procedure, allowing you to create sophisticated find and change routines.

When you specify a PDS, File Manager allows you to search for or change a string across a list of selected members within the PDS. When working with compressed non-VSAM extended format data sets (compressed PSE data sets), the CHANGE command is not supported, however, the FIND command can be used.

The overall process for using this utility is to:

1. Select the Find/Change Utility option from the Utility Functions panel. The Find/Change Utility panel (shown in Figure 178 on page 574) is displayed.
2. Build a list of PDS members to be processed (or specify a single VSAM or sequential data set).

3. Specify the number of records to be searched in each data set or PDS(E) member.
4. Specify where a listing of the results of your changes is to be stored.
5. Select your processing options.
6. Enter a FIND or CHANGE command in the Command line or specify a REXX procedure to be used.
7. Press Enter to process the panel and command.

   Note: If your PDS data set list has over 1,000 members or your sequential data set has over 10,000 records, you will see a progress indicator in the upper right corner of the screen, displaying the count of members or records processed in the FIND or CHANGE action. For example:

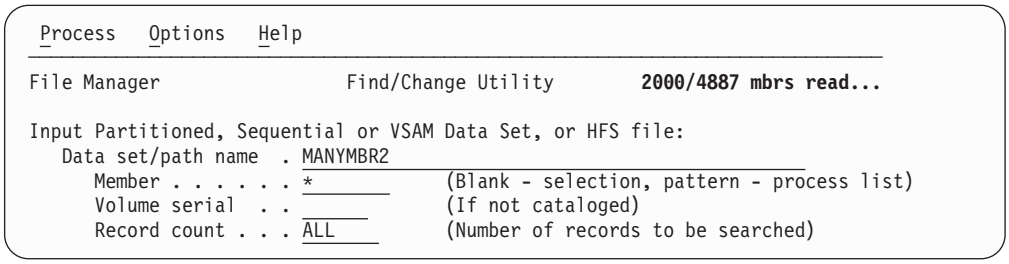


Figure 75. Progress indicator showing PDS members

   When the command has finished running, the Listing data set, showing the results of your command, is displayed in a Browse session.
8. Press F3 to return to the Find/Change Utility panel.

This tasks described in this section are:

Building your member list for processing

The Find/Change Utility is particularly useful when you are working with partitioned data sets. You can specify that either *all* members or only *selected* members of a PDS are searched by the FIND or CHANGE command you enter on the Command line.

Note: The selection described in this section is not applicable when the batch execution option is selected.

Perform either of these actions

- Specify the sequential data set, VSAM data set, or PDS member that contains the data you want to search, using a combination of the **Data set name**, **Member**, and **Volume serial** entry fields.

  If you leave the **Member** entry field blank, File Manager displays the F/Change Member Selection panel showing all members of the PDS:

  – Select the members you want either by typing an "S" in the **Cmd** field, or with the SELECT primary command, and press Enter.

– Press F3 (Exit).

The Find/Change Utility panel is redisplayed showing the selected members in the **Process List**.

You can also select members by entering a member name pattern in the **Member** entry field.

For example, to select all members whose name starts with "TRAN":

1. Enter `TRAN*` in the **Member** entry field
2. Press Enter

The Find/Change Utility panel shows the relevant members in the **Process List**.

- To select a range of members based on selection criteria, specify the data set name or name pattern for a PDS(E) data set, optionally the member name or pattern, and select the **Advanced member selection** option:
  – When you press Enter, File Manager displays the Advanced Member Selection panel.

    Note: If you enter details in the **Member** field and select the **Advanced member selection** option, File Manager populates the **Member name** field on the subsequent Advanced Member Selection panel with the same value.
  – Specify the selection criteria to select the members you want to print and press Enter.

    If you have left the **Member name** field on the Advanced Member Selection panel blank or have entered an asterisk (*) or a mask, File Manager displays the F/Change Member Selection panel. In this case, select the members you want to search (either by typing an "S" in the **Cmd** field, or with the SELECT primary command) and press Enter. The selected members are shown in the **Process List** on the Find/Change Utility panel.

Related topics

"Find/Change Utility panel" on page 574
"Specifying a data set and a member name" on page 16
"Member Selection panel" on page 600
"FCH (Find/Change)" on page 1015
"Selecting a range of PDS(E) members" on page 34
"Advanced Member Selection panel" on page 439

Refining the list of selected members

You can further refine the list of selected members in the **Process List** by using the FIND command to search for a specific string. The **Process List** is then reduced to only those members that contain the specified string.

For example, having created a **Process List** of members whose name start with "TRAN", to refine the list to members that contain the string "salesperson-code":

1. Enter `FIND SALESPERSON-CODE` on the Command line.
2. Press Enter. File Manager displays a panel showing the records within each member that contain the string "salesperson-code" Figure 76 on page 273.

```
   Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    FMNUSER.SRCHFOR.LIST                      Line 00000000 Col 001 080
******************************** Top of Data ********************************
IBM File Manager for z/OS
Find/Change Listing DSN:FMNUSER.FMOS390.COPY

 TRANREC2                  ---------- STRING(S) FOUND ----------
Record Number
     18            05 salesperson-code           Pic 9(4).

  -- Find/Change summary section --
  Records found: 1 Records processed: 65
  Members w/recs: 1 Members wo/recs: 1
  Search cols: 1:80 Longest line: 80
  Edit options in effect: CAPS:OFF
  Search argument: >SALESPERSON-CODE<
******************************* Bottom of Data ********************************
 Command ===> _____________________________________________ Scroll ===> PAGE_
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
```

Figure 76. Find/Change Utility: example of results from FIND command

3. Press F3 (Exit). The Find/Change Utility panel is redisplayed showing the refined list of members in the **Process List**.

Alternatively, you can use the FINDNOT command to reduce your list of members to only those that DO NOT contain a string. The FINDNOT command can only be used for a PDS or PDS/E data set.

The output listing displayed after the FINDNOT command is issued, is similar to that displayed for the equivalent FIND command, except that:

- The summary section contains the list of members that were excluded from the process list as a result of the search string being found.
- When LIST=LONG is specified, only the first occurrence of the specified string in each member is displayed.
- The summary section contains an extra statement indicating that the search was initiated by the FINDNOT command and showing the count value of the number of members NOT satisfying the search criteria.

You can also exclude members from the **Process List** by entering S in the **Sel** field for the members you want to exclude. You can toggle between excluding a member or not by entering S in the **Sel** field.

To refresh the **Process List** to its original state (that is, the list of members as determined by the combination of the **Data set name**, **Member**, and **Volume serial** entry fields), press the Refresh function key (F5).

When you have generated the desired **Process List**, you can use the FIND or CHANGE primary commands to find or alter data in the selected members.

Note: You can restore your refined Process List to the full list (as specified by the data set Input fields), by entering the REFRESH command (F5).

Related topics

"FIND/FX primary command" on page 769
"FINDNOT primary command" on page 777

Specifying where your results are stored

Use the **Listing data set** entry field to specify the data set in which the Find/Change Utility results are stored. If you leave this field blank, the output is stored in the data set *userid*.SRCHFOR.LIST where *userid* is your ISPF user ID.

The listing data set can be an existing data set (in which case it is overwritten). If it is not, it is allocated for you.

Specifying your processing options

Once you have your list of data sets or members built up, you can specify the processing options that you would like to use when performing your find or change operations.

Choose among the following options:

JCL source format
Select this option when your data set contains JCL and you want that JCL syntax to be preserved when making changes.

The following JCL statements are supported:
- DD
- EXEC
- JCLLIB
- JOB
- OUTPUT
- PROC
- SET
- XMIT

All other statement types are treated as data records.

When File Manager recognizes one of the supported statement types, it attempts to process the CHANGE command in such a way as to preserve the JCL syntax. For the eligible JCL statements, changes in a record that would result in the text being longer than 255 characters before reformatting (due to the new string being longer than the old string) within a JCL statement, are not made.

The file must be non-VSAM and have a fixed record length of 80.

Use REXX proc
You can use this option to specify a REXX procedure that manipulates your records in more complex ways than by issuing FIND and CHANGE commands within the panel.

REXX no update
Select this option when your REXX procedure performs no updates to the FCH data set. If selected, it forces the allocation of the data set as input only. All updates to the data are ignored.

Use I/O exit
Select this option when you want to specify a user I/O exit for compressed or encrypted data sets.

Immediate change
Select this option when you intend to use the CHANGE command, but want the input data set to be updated immediately (without displaying the changes in the listing data set).

Batch execution
Select this option when you want File Manager to create JCL that reflects the FIND or CHANGE command that you entered on the Command line. The JCL is presented in an Edit session which you can edit before submitting.

Batch execution restricts the member selection to the pattern specified in the member field. Batch execution does not produce a pop-up selection panel for member selection. If you leave the member field blank, an asterisk (*) is substituted in the JCL generated. For more information, see "FCH (Find/Change)" on page 1015.

Stats off
When selected, this option ensures that the ISPF statistics for PDS or PDSE members, if present, are not updated. Selecting this option improves performance when processing PDS or PDSE data sets with many members.

Directory integrity
Forces an override of the default PDS(E) member processing method which allows for faster PDS directory access.

This option has significant performance impact. When selected, the members are processed in a way which allows concurrent directory updates as File Manager accesses the members using current directory information.

When not selected, the member processing is performed faster, but may be affected by PDS(E) directory updates, possibly causing I/O errors if the data set is updated concurrently.

Listing Option
Use this option to choose the format of the output report.

1 A full report, including each record found or changed.

2 A summary report providing totals for records processed and strings found and changed.

ISPF Packing
Provided that the output data set is a sequential, PDS or PDSE file, an I/O exit routine is not used and the DISP is set to OLD, select one of these options to control the copy behavior when processing data that is in ISPF PACK format.

1. Asis
Select this option when you want File Manager to write the output in the same format (packed or unpacked) as the input records. If the input data is packed, it is unpacked for the processing operation, and then written out in ISPF PACK format.

2. Unpack
Select this option when you want File Manager to write the output in unpacked format, regardless of the format of the input records. If the input data is packed, it is unpacked for the processing operation, and then written out without the ISPF PACK format.

3. Pack
Select this option when you want File Manager to write the output in ISPF PACK format, regardless of the format of the input records. If the input data is packed, it is unpacked for the processing operation, and then written out in the ISPF PACK format.

4. None

Select this option when you do not want File Manager to check for packed records or unpack records for processing. If the input data is packed, any processing actions operate on the records in their packed format.

This is the only option allowed when an I/O exit has been specified, or when the data set is not a sequential file or PDS(E) member.

5. Skip

Select this option when you want to halt the processing actions if the input data is packed.

Specifying multiple arguments for FIND

You can specify multiple arguments for the FIND command, by using OR clauses or AND clauses. You cannot use both AND and OR clauses in the same search. If you specify one or more OR clauses, then a record is selected when any of the arguments are found in the record. If you specify one or more AND clauses, then a record is selected only if all of the arguments are found in the record. The maximum number of arguments that you can specify is 16.

When you use a FIND command in batch, you may need to use more than one line to specify multiple arguments. To indicate that a line is continued, ensure the last item on the line is a blank-delimited comma.

The following command finds records containing either of the strings “GROSS” or “NET”:

```
FIND GROSS | NET
```

The following command finds only records containing all of the strings“HAPPY DAYS”,“HERE”, and “AGAIN”:

```
FIND 'HAPPY DAYS' AND HERE AND AGAIN
```

Related topics

- “FIND/FX primary command” on page 769

Using the CHANGE command in the Find/Change Utility

The CHANGE command operates on the data set or data set members selected in the Find/Change Utility. The CHANGE command or any of its abbreviations is entered on the Command line, together with the desired parameters. To find and replace long strings that do not fit on the Command line, enter the CHANGE primary command (or one of its abbreviations, such as C) *with no parameters*, or enter the CX command on the Command line with no parameters. This displays the Extended Command Entry panel, in which you can enter long strings and the CHANGE command parameters.

You can limit the effects of the change command by issuing the BOUNDS command before issuing the CHANGE command, or by using one or more of the following parameters:

col1 **and** *col2*

Col1 is the first column to be included in the range of columns to be searched. It must be greater than or equal to 1, and less than or equal to the maximum record length. *Col2* specifies the last column to be included in the range of columns to be searched.

MAXINREC(*n*)
The MAXINREC(*n*) parameter allows you to specify the maximum number of **changes** that can be made within a **single record**.

MAXRECS(*n*)
The MAXRECS(*n*) parameter allows you to specify the maximum number of **records** that can be changed within a **single data set or PDS member**.

FIRST(*n*)
The FIRST(*n*) parameter allows you to specify the maximum number of **total changes** that can be performed within a **single data set or PDS member**.

Suppose you want to change the string "SALESPERSON-CODE" to "SALESPERSON-NUMB" in certain members of a PDS.

1. Select the members you want to search.
2. Enter `CHANGE SALESPERSON-CODE SALESPERSON-NUMB`. File Manager displays a panel showing the changed records (only one in this case) as shown in Figure 77.

```
  Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    FMNUSER.SRCHFOR.LIST                      Line 00000000 Col 001 080
******************************** Top of Data **********************************
IBM File Manager for
Find/Change Listing DSN:FMNUSER.FMOS390.COPY

 TRANREC2                  ---------- STRING(S) FOUND ----------
Record Number
     18            05 salesperson-numb           Pic 9(4).

  -- Find/Change summary section --
  Records found: 1 Records processed: 33
  Members w/recs: 1 Members wo/recs: 0
  Search cols: 1:80 Longest line: 80
  Edit options in effect: CAPS:OFF
                Count  Value
  Found   :         1  >SALESPERSON-CODE<
  Changed :         1  >SALESPERSON-numb<
 Command ===>                                                Scroll ===> PAGE_
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
```

Figure 77. Find/Change Utility: example of results from CHANGE command

Note: On the Find/Change Utility report, the record number can have one of these prefixes:

K Indicates the change involved a key.

X Indicates that although it was eligible to be changed, the change could not be performed due to record length restrictions.

KX Indicates both of the above.

3. Press F3 (Exit)
   The PDS Find/Change pop-up window appears.
4. Press Enter to save or F3/End to cancel changes.

Related topics
"CHANGE/CX primary command" on page 748

Specifying context in the output listing

There are times when you might want to view the context of a found or changed string, in the output listing. To do this, you can enter the VCONTEXT command, or set its value as an option, on the Find/Change Utility panel, with parameters to specify how many records before and after the found or changed string should be displayed. For example, if you entered `VCON 2 2`, each found or change string is displayed with the 2 prior and subsequent records. This can be particularly useful when searching data set members containing JCL.

The VCONTEXT command affects all output displays for the remainder of the current session. It can be changed within a session or reset to zero by entering the command with no parameters.

Note: The VCONTEXT command only affects the output display when the **Listing Option** is set to **Long**.

```
******************************** Top of Data ********************************
IBM File Manager for z/OS
Find/Change Listing DSN: USERID.MANYMBR2

 M100                        ----- STRING(S) FOUND/CHANGED -----
Record Number
      3  //*ROUTE PRINT STLVM20.gregcz
      4  //*******************************************************************//
      5  //*CLEANUP  EXEC PGM=IEFBR14
      6  //*OLDLISTS DD DISP=(MOD,DELETE),
      7  //*             DSN=ECHAS.PLX370.TEST.LISTPLX,
     83  //UNIDATA  DD SYSOUT=*
     84  //*******************************************************************//
     85  //TESTASM  EXEC PGM=IEFBR14 ASMA90,COND=(9,LE,COMPILE)
     86  //*TESTASM  EXEC PGM=ASMA90,COND=(9,LE,COMPILE)
     87  //*TESTASM  EXEC PGM=ASMA90,PARM='ADATA',COND=(9,LE,COMPILE)

  -- Find/Change summary section --
  Records found: 2 Records processed: 111
  ISPF packing option: ASIS
  Search cols: 1:80 Longest line: 80
  Edit options in effect: CAPS:OFF

                Count  Value
  Found   :         2  >EXEC PGM=IEFBR14<
------------------------------------------------------------------------------
```

Figure 78. Sample output listing with VCONTEXT 2 2 specified

Related topics

"VCONTEXT primary command" on page 827

Changing data with a REXX procedure

You can specify an existing REXX procedure, or create a new one, to further enhance the way in which data is searched.

The find/change procedure is an alternative way to specify find and change commands that provides the flexibility of a REXX procedure. For Find/Change Utility, the REXX procedure is mutually exclusive to the primary commands CHANGE, BOUNDS, and CAPS.

However, you can use the FIND primary command to refine the selection list for the procedure. To do this, do not select the procedure until you have produced the

list of members to be processed. You can deselect a record by executing an exit DROP statement. The Find/Change Utility report prefixes each record number *selected* by the REXX procedure with an "S", each record *changed* by the REXX procedure with a "C", and each record *added* by the REXX procedure with a "+".

For general information about using REXX procedures with File Manager options, see Chapter 13, "Enhancing File Manager processing," on page 405.

Working with files or members containing JCL

When using the File Manager Utilities DSC or FCH with a file containing JCL and the JCL Source format option has been selected, the following regime is applied to maintain integrity:

- The rules only apply to legitimate JCL cards (// cards but not //*):
  - When a change would cause truncation due to the insertion of characters, the card is split at a suitable position. If possible, File Manager inserts the part that was cut into the next card; otherwise, an extra card is be created.
  - When a change causes the removal of data, the card is changed to a comment card.
  - In both of the above, File Manager attempts to maintain the integrity of matching quotes, matching brackets, and commas between parameters.
- The rules do not apply to:
  - Comment (//*) cards.
  - Imbedded comments at the end of a legitimate JCL card
  - Non-JCL cards. For example, data cards.
  - JCL cards contained within a DD DATA construct as these are considered to be data cards.

The rules detailed above only apply to cards that are changed. If a card is not changed then the syntax is *not* checked.

Any JCL file that is referenced is assumed to have valid syntax. The integrity checking performed by File Manager should not be seen as a method of remedying invalid JCL.

When reformatting JCL cards, File Manager takes into account the following:

- The first card of a logical JCL statement contains one of the following statement types:

  CNTL
  COMMAND
  DD
  ENDCNTL
  EXEC
  IF
  THEN
  ELSE
  ENDIF
  INCLUDE
  JCLLIB
  JOB
  OUTPUT
  PEND

PROC
SET
XMIT

- Each logical JCL card may be comprised of one or more physical JCL cards.
- Comment (//*) cards are not candidates for reformatting.
- Data cards (no "//" in columns 1-2) are not candidates for reformatting.
- Each logical JCL card may have imbedded comment (//*) cards which are also not candidates for reformatting.
- The logical JCL statement line consists of the logical JCL statement itself (name, operation, and parameter fields), with continuation syntax removed. The logical JCL statement is delimited with a x'FF'. Any comments will then be appended with each comment being separated by a x'FF' delimiter.
- Each physical JCL card is comprised of 4 parts:

  Name part
  Contains the contents of column 1 up to the last character before the *parameter part*, including the card type. For example, DD, JOB, and so on. This part may also only contain "//" as in the case of a continuation line.

  Parameter part
  Contains the contents of the first character after the *name part*, including any blanks after the last character of the *parameter part*.

  Embedded comment part
  This part is optional. It contains the contents of the first non-blank character after the *parameter part* up to column 71 inclusive. Column 72 is a special column and as such is not considered a candidate for reformatting.

  Sequence number part **(including continuation character)**
  Contains the contents of columns 72 to 80.

When building a new set of physical JCL cards from a logical JCL statement, the following processing takes place:

- If the logical JCL statement line is unchanged, File Manager does not change any part of the associated physical records.
- If the logical JCL statement line content changes in any way, then File Manager reflows the logical JCL contents into the associated physical records.

  Note: It is possible for a logical JCL statement change to require reflowing even if the old and new strings are identical in length-- the changed logical JCL statement might need to be broken into physical records differently.

  – File Manager attempts to maintain the contents of the existing physical line (columns 73–80) and the existing JCL statement comment fields. However:
    - File Manager does not attempt to associate any part of a logical JCL statement with the related physical line columns (columns 73–80), or with the JCL statement comment fields. The physical columns (columns 73–80), the JCL statement comment fields, and logical lines are treated independently of each other.
    - JCL statement comment fields may be truncated.
    - File Manager may remove redundant blanks from, and otherwise reformat, the physical JCL lines.
    - If a logical JCL statement is changed and it requires fewer physical records, File Manager turns the "extra" physical records at the end into JCL

comment statements, which start with //* and contain blanks in the JCL statement area. Data in the comment record columns 73–80 remains unchanged from the original non-comment record data.

– If an entire sub-parameter is removed as a result of a change, and this was the only information on a card, the card is changed to be a comment card.

Additional rules which pertain to JCL statements (those starting with //) passed to a REXX procedure:

- The REXX input and output buffers (INREC and OUTREC) contain the logical JCL statement, including the // at the front.
- The REXX procedure processing can only access and affect the logical JCL statement.
- File Manager ignores any changes made to the // characters in bytes 1 and 2 of a logical JCL statement in an output buffer (OUTREC).

Comparing data sets

When you are using File Manager under ISPF, you can use the Compare Utility (option 3.11) to compare data from one data set with data in a second data set; when you are programming a batch job, REXX procedure or TSO clist, you can use the equivalent function, DSM (Data Set Compare). Typically, the first data set contains the original version of some data (the “Old” data set), and the second data set contains an updated version of the data (the “New” data set).

Note: The terms “Old” and “New” do not imply any significance in regard to the dates on which the data sets or templates were created or modified. Any data set or template can be specified as either “Old” or “New”.

Whether you are using the panel or the function, you can compare data without using templates, or by specifying a template to format either the “Old” data set (the “Old” template) or the “New” data set (the “New” template), or for both “Old” and “New” data sets, in order of increasing flexibility.

Record selection can be performed at the record level using the **Start key**, **Skip count**, and **Compare count** fields, or at the field level using conditional expressions defined in the template, or at a combination of both levels. The **Start key** (or slot) and **Skip count** fields are honored before field-level selection takes place. The **Number of differences to report** option allows you to limit the number of differences reported by the compare operation.

To perform field-level selection you must specify a template or copybook. If field-level selection is in effect, the compare count is not met until the specified number of records have been selected.

Note: The subset of data set records defined by the total of all record and field-level selection criteria is referred to as the “compare” set.

You can use the Compare Utility to:

- Compare data from any supported data set to data in any other supported data set.
- Select the records to be compared (the compare set), using the start key (VSAM only), skip and copy count fields, a conditional expression defined in a template, or a combination of all of these.

- Perform a field-level comparison. By using an “Old” copybook or template with a “New” copybook or template you can compare selected fields, with the result of the comparison reflecting the types of data in the fields.
- Restrict the number of differences reported with the Compare Utility option, **Number of differences to report**.
- Create output data sets containing records identified as inserted, deleted, old and new changed records, and old and new matched records.
- Perform load module comparison. Load module and CSECT information from both "Old" and "New" versions of the module is extracted and compared. By selecting from various compare criteria, you can see differences between specific attributes of load modules like load module size, link date, CSECT names, and compilers used.

The compare operation can be performed in the background (batch execution) or in the foreground.

With or without templates, you can compare records where the “Old” and “New” data sets have different:
- Record formats
- Record lengths
- Block sizes

Comparing data sets without using a template

A “simple” compare: all data in the “Old” data set is compared with that in the “New” data set, with only the limited additional options that you can specify on the Compare Utility (option 3.11) panel (for example, you can specify the number of records to be compared).

To perform a compare between data sets:

1. From the Primary Options menu, select the Compare Utility (option 3.11).

   The Compare Utility - “Old” data set entry panel (Figure 134 on page 472) is displayed.
2. Specify the “Old” data set information.
3. Set your record-level selection, using the following fields:

   Start key
   If required, specify a **Start key** for the data (VSAM only). The comparison starts at the first record with a key or slot number greater than or equal to the specified value. If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. Keys may also be in hexadecimal format (e.g. X'0102').

   Skip count
   If required, specify a **Skip Count** for the data. The comparison starts after skipping the number of records specified in the field.

   Compare count
   Specify the number of records to be compared. Enter ALL to compare all records.

   Note: The Start key and Skip count fields are mutually exclusive.
4. Select **3. None** in the **Copybook/template usage** field.

   Note: When this field is set to None, data in the **Copybook or Template Data set name** and **Member** fields is ignored.

5. If you want to run this process as a batch job, select the **Batch execution** option.
6. If required, select the **Use I/O exit** option to run a user I/O exit that processes the "Old" (input) data set in conjunction with the compare action. If selected, specify the exit name in the adjacent field.

   Note: This option is only available when File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field in the Set System Processing Options panel is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of **Use I/O exit**.
7. Press Enter. The Compare Utility - "New" data set entry panel is displayed.
8. Specify the "New" data set, **Start key**, **Skip count**, **Compare count** and **Use I/O exit** fields as required for the "New" data set and press Enter.

   The Compare Utility - Options panel is displayed.
9. Set the **Compare type** to **1. Record** (you cannot use the Formatted option unless you have specified a template for both the "Old" and "New" data sets).
10. Select your **Listing type** from the available options.
11. If you have selected a listing type of **Long**, you can choose to exclude certain categories of records from the output report by selecting one or more of the **Long Report** options:
    - To exclude inserted records from the report, select **Inserted**.
    - To exclude deleted records from the report, select **Deleted**.
    - To exclude changed records from the report, select **Changed**.
    - To exclude matched records from the report, select **Matched**.
12. If required, select the **Clear print data set** option. This option overrides your current Print DISP setting but has no effect if File Manager's print output is not directed to a data set.
13. If required, select the **Create result data sets** option to create one or more output data sets containing the results of the compare process. (You specify the output data sets on the subsequent Output Data Sets panel.)

    Note: If you have selected the synchronization option, **Read-ahead**, this option is ignored.
14. Select your **Listing options** from the list.
15. If your data is in a sequential data set or PDS or PDSE member, is in ISPF PACK format and you have not specified an I/O exit, you can select an ISPF Packing option.

    1. Unpack
    Select this option when you want File Manager to unpack the data before any user processing or comparison occurs.

    2. None
    Select this option when you do not want File Manager to detect the ISPF PACK format and to operate on any packed data in its packed format.

    This is the only valid option when your data is not a sequential or PDS data set, or when the Use I/O exit option has been used.

    3. Skip
    Select this option when you want File Manager to detect data in ISPF PACK format and stop any processing or comparison for the data set or member.

16. If required, in **Number of differences to report**, specify the maximum number of differences (found in the compare operation) that you want reported.
17. Note: When you compare data sets without using a template, the **Template Reporting Options** on the Compare Utility - Options panel have no effect.
18. Press Enter.

    If you selected Read Ahead or Keyed Synchronization, the relevant panel for that option is displayed. Complete the details as instructed above and press Enter again to continue.

    The comparison operation is performed and the results are displayed on the screen.

    If you selected “Batch Execution”, the result is an ISPF Edit session containing the generated JCL and File Manager control statements to execute the requested comparison.

    If you did not select the “Batch Execution” Processing Option, the result is a report. This might be directed to File Manager's print data set, or elsewhere, depending upon your Print PRINTOUT setting.

    Note: For best viewing when you are comparing wide files using the **Wide listing** and **Show hex chars** options, set your PRINTOUT to SYSOUT=C. This prevents the report lines from wrapping around on the screen.
19. If the report has been displayed online, press the Exit function key (F3) to return to the Compare Utility - “Old” Data Set Entry panel.

Related topics
“Compare Utility: “Old” and “New” panels” on page 471
“Compare Utility: Options panel” on page 479
“Setting your Print Processing Options” on page 299

Comparing data sets using templates

When you choose to use templates in your comparisons, you can specify an “Old” template only, a “New” template only or both “Old” and “New” templates.

"Old" template only or "New" template only

This excludes records not-selected in either the “Old” or the “New” data set.

During the compare, the Compare Utility uses the “Old” or “New” template to provide information about the selected record types, record identification criteria and record selection criteria for the “Old” and “New” data sets. Any field selection and field resequencing information in the templates is ignored, however, you can specify fields to be used in a multi-segment key that can be used with the Keyed Synchronization comparison option.

You determine which records are selected or excluded by editing the template and:
- Selecting record types
- Specifying record identification criteria
- Specifying record selection criteria

Not-selected records:
- do not match a record type in the template, due to either record identification criteria or record length;
- match a record type, but belong to an unselected record type; or
- do not meet the record selection criteria for the record type.

"Old" and "New" templates (based on the same copybook or dynamic template structure)

This allows you to exclude records from both the “Old” and the “New” data sets, based upon the template information, and allows you to specify which fields in the “Old” data set are compared with which fields in the “New” data set.

During the compare, the Compare Utility uses the “Old” and “New” templates to provide information about the selected record types, record identification criteria and record selection criteria.

You can also specify key segment sequencing to create a multi-segment key that can be used with the Keyed Synchronization comparison option.

If a formatted comparison is performed, the record structure, field output width and field heading attributes in the “Old” and “New” templates are also used. Field mapping information is taken from the “New” template and only fields that have been mapped are compared. A field in the “Old” template can be mapped to the same field in the “New” template (this is the default), to a different field or unmapped.

"Old" and "New" templates (based on different copybook or dynamic template structures)

This allows you to exclude records from both the “Old” and “New” data sets and allows the “Old” and “New” data sets to have different record structures, according to the differences in their copybook or dynamic templates.

You may have existing templates for both data sets, or you might copy a template describing one of the data sets and edit it to describe the other data set.

The table below summarizes how the Compare Utility uses information in the “Old” and “New” templates:

The Compare Utility determines...	Using this information...	In this template...	
		Old	New
Which records are compared	Selected record types Record identification criteria Record selection criteria	✓	✓
Which fields are compared, and with what	Field mapping		✓

If a formatted comparison is performed then, by default, the fields in the “Old” template are mapped to fields in the “New” template that have the same name. You can edit the field mapping in the “New” template, and either delete the mapping for a field (so that it is not compared), or specify which field in the “Old” template is mapped to a field in the “New” template.

To perform a compare using templates:

1. From the Primary Options menu, select Compare Utility (option 3.11).

   The Compare Utility - “Old” Data Set Entry panel is displayed.
2. Specify the “Old” data set information and, optionally, the “Old” template information.
   - To use a template, choose any one of these methods:

  - Specify an existing dynamic template, an existing copybook template or an existing copybook on which you want to base a new template and choose **1. Above** from the Processing Options.
  - Choose **2. Previous** from the Processing Options. The template most recently associated with the "Old" data set is used and the contents of the "Old" template fields are ignored.
  - Choose **4. Create Dynamic** from the Processing Options. The contents of the "Old" template fields are ignored and you are asked to create the new template when Enter is pressed.
- If you want to map fields from the "Old" to the "New" data set, you *must* specify an "Old" template.
- If you want to edit the "Old" template information before using it, select the **Edit template** option.
- Press Enter.
  - If Processing Option 1 or 2 was selected AND the **Edit template** option was NOT selected, the Compare Utility - "New" Data Set Entry panel is displayed.
  - If Processing Option 1 or 2 was selected AND the **Edit template** option WAS selected, the panel appropriate for editing the template is displayed. Make your record identification, record selection, field attribute and key sequencing choices and press the Exit function key (F3) to return to the "Old" data set panel. Press Enter again to move to the "New" data set panel using the modified template.
  - If Processing Option 4 was selected, the Dynamic Template panel is displayed. Create and save your new dynamic template. If you want to include key sequence information, select option **1 Edit template field attributes and selection** and make your selections in the Field Selection/Edit panel. When you have finished creating your template, press the Exit function key (F3) to move to the "New" data set panel.

  Note: If you do not want to save the changes to your template, you can use the RunTemp function key (F6) in place of the Exit function key (F3).
- In the Compare Utility - "New" Data Set Entry panel, specify the "New" data set information and, optionally, the "New" template information.
  - If you want to map fields from the "Old" to the "New" data set, you *must* specify an "New" template.
  - If you want to edit the "New" template information before using it, select the **Edit template** option.
- Press Enter.
  - If Processing Options 1 or 2 were selected and the **Edit template** option was NOT selected, the Compare Utility - Options panel is displayed.
  - If Processing Options 1 or 2 were selected and the **Edit template** option WAS selected, the appropriate panel for editing the template is displayed. Make your record identification, record selection, field attribute and key sequencing choices, and press the Exit function key (F3) to return to the "New" data set panel. Press Enter again to move to the Options panel using the modified template.
  - If Processing Option 4 was selected, the Dynamic Template panel is displayed. Create and save your new dynamic template. If you want to include key sequence information, select option **1 Edit template field attributes and selection** and make your selections in the Field

Selection/Edit panel. When you have finished creating your template, press the Exit function key (F3) to move to the Options panel.

Note: If you do not want to save the changes to your template, you can use the RunTemp function key (F6) in place of the Exit function key (F3).

- Select a combination of Compare, Processing and Listing options. These options apply whether the comparison type is record or formatted, except that for formatted comparisons the word "record" refers only to those parts of the record specified in the mapping.
  - If you want to use the field mapping information within the “New” template, you *must* choose the **Formatted** Compare Type.
  - If you want to edit the field mapping information within the “New” template, select the **Edit template mapping** option.
  - If you select Keyed synchronization, you can:
    - Specify the location, length and data type of your key segments in the Keyed Synchronization Settings panel that is displayed after the Compare Options panel. If a data type is not explicitly specified, type AN is used.
    - Use a template to specify fields to be included in the key and the sequence in which they are concatenated (known as key sequence information). If you want to use an intrinsic key as well as your own key sequence information, you must start the key sequence at 2 or higher. If your key sequence information starts at 1, any intrinsic key information is ignored for that data set.
- Select your **Template Reporting Options** from the list:

  Show template layouts
  Select this option to report template layout information. You must also have set the **Compare type** to **2. Formatted**.

  Show template criteria
  Select this option to report template criteria information. A terse layout report is also generated, containing only the fields referred to in the criteria expressions.

  Show mapped fields
  Select this option to report mapped template fields. You must also have set the **Compare type** to **2. Formatted**.

  Show unmapped fields
  Select this option to report unmapped template fields. You must also have set the **Compare type** to **2. Formatted**.

  Show array elements
  If field information is reported (that is, if you have selected any of the first four **Template Reporting Options**), select this option to report all occurrences of any array elements. Note that for COBOL OCCURS DEPENDING ON tables, the maximum occurrences are reported, and for PL/1 REFER arrays, the minimum occurrences are reported.

  Show start/end as hex offset
  If field information is reported (that is, if you have selected any of the first four **Template Reporting Options**), select this option to report all field start and end positions as hexadecimal offsets.

  Show length in hex
  If field information is reported (that is, if you have selected any of

the first four **Template Reporting Options**), select this option to report all field lengths in hexadecimal.

- Press Enter.
  - If you have not selected the **Edit template mapping** option, the comparison operation is performed, using the current settings and template information. The result is either a Compare Report or a batch job (displayed in the Edit panel).

    Note: If no fields are selected in the saved or default-generated field mapping, the **Edit template mapping** option is forced.
  - If you have selected the **Edit template mapping** option, the appropriate panel for editing the field mapping is displayed. Make mapping choices and press the Exit function key (F3) to return to the Options panel. Press Enter again to perform the compare with the modified template.
- When you have finished viewing the report or editing the batch JCL, press the Exit function key (F3) to return to the "Old" data panel.

Related topics

"Compare Utility: Options panel" on page 479
"Editing a template" on page 156
"Creating dynamic templates" on page 152
"Mapping fields between templates" on page 184

Record synchronization

File Manager synchronizes records for comparison in several ways.

1. One-to-one

When you select this option, the compare sets are assumed to contain records in a corresponding sequence.

Non-matching records are regarded as changed records. No attempt is made to resynchronize the compare sets when non-matching records are encountered, and comparison proceeds with the next record from each compare set. Trailing records in the "New" compare set are regarded as unpaired insertions, and trailing records in the "Old" compare set are regarded as unpaired deletions.

2. Read-ahead

The data sets are assumed to contain roughly corresponding records.

Specification requirements

When read-ahead synchronization is requested, you specify a limit and a length for read-ahead synchronization processing.

The read-ahead synchronization limit controls how far ahead File Manager looks for matching records when a pair of non-matching records is found. You should try to specify a read-ahead limit a little larger than the maximum number of contiguous non-matching records you expect to be encountered in the data sets.

The read-ahead synchronization length determines how many contiguous matching records must be found for synchronization to be recognized. If you are comparing "data" records, where false matches due to blank records or other commonly repeated records are unlikely, you should specify a small read-ahead synchronization length, such as the default value of 1. If you are comparing "text" or "source" records, where false

matches due to blank records or other commonly repeated records might be common, you should specify a read-ahead synchronization length greater than one.

How the records are synchronized

When non-matching records are encountered, File Manager attempts to resynchronize the data sets by looking ahead in each data set by up to the number of records specified by the read-ahead limit. If the resynchronization is successful then the non-matching records are regarded as some combination of paired and unpaired insertions and deletions, depending on where the resynchronization occurred. If the resynchronization fails, then all of the "old" and "new" records from the first mismatch up to the resynchronization limit are regarded as paired insertions and deletions.

3. Keyed

When you select this option, the compare sets are assumed to be sequenced by a key, which is comprised of one or several segments of the record (up to sixteen). When you press Enter to process this panel (after having made your selections for the remaining options) the Keyed Synchronization Settings panel is displayed, in which you must specify at least one key segment (and up to sixteen key segments). Any existing key segment information, such as that drawn from an intrinsically keyed data set, is displayed in the panel. You can edit or modify these details.

When you have not used a template, the following applies:

- If your data set contains an intrinsic key, as in KSDS data sets, File Manager automatically attempts to use this key as the first (or only) segment in the synchronization key.
- You can specify the location, length and data type of your key segments in the Keyed Synchronization Settings panel that is displayed after the Compare Options panel. If a data type is not explicitly specified, type AN is used.

When you define multiple key segments, each segment is checked in the specified key segment order. If an alphanumeric key field falls partly or wholly outside of a record so that the key is truncated, the key comparison is performed as though the key were padded with trailing hex zeros. For other key types, any truncated key after the first record results in a sequence error as well as a truncation error.

If the keys do not match, records are read from the compare set with the lower key until a matching or greater key is found. Non-matching records with matching keys are regarded as changed records (tallied as paired changes in the comparison summary). Records with non-matching keys are regarded as unpaired insertions and deletions. Records with keys that are out of sequence are flagged with a plus sign (+) in the comparison report. Records with truncated keys are flagged with a minus sign (-) in the comparison report. Records with keys that are both out of sequence and truncated are flagged with an asterisk (*) in the comparison report.

If you opt to use intrinsic keys for both your “Old” and “New” data sets and there is a length mismatch, File Manager uses the shorter of the two keys as the segment length. If there is any other type or length mismatch between key segments defined for the “Old” and “New” data sets, the Keyed Synchronization Settings panel contains additional columns that indicate the conflict.

You can ignore the conflict and process the panel using the values shown for the “Old” data set, or you can edit the “Old” key segment information and then continue.

You can also use record selection to affect the apparent synchronization of your data sets. As a simple example, imagine you have copied a file and inserted ten new records at the start. To verify that you have not corrupted any of the existing records you might compare the original data set with the new data set, and request a read-ahead comparison with a limit of ten, which would show the added records as unpaired insertions. Alternatively, you might request a one-to-one comparison with a skip count of ten on the “New” data set, which would create a New compare set that ignores the added records completely.

Creating key sequencing for use in a multi-segment key

When you use the keyed synchronization comparison option, you can specify up to 16 segments of record information to be used as the key. A key segment is comprised of the key's location in the “Old” and “New” data sets and the key's length and data type. These values are built using one of the following sources or a combination of these elements:

- The intrinsic data set keys (where the data set is keyed, for example, VSAM KSDS).
- The key segment sequence information stored in your templates.
- The key location and length information supplied in the Keyed Synchronization Settings panel. This panel is displayed whenever you choose the keyed synchronization option.

To create the key segment sequence information in your template:

1. Display your template in the Field Selection/Edit panel.

   When this is done as part of editing or creating a template within the compare process, the panel shows the **Key** field in place of the **Rdf** field. When done as part of editing or creating a template from any other location within File Manager, you need to enter the KEYFLDS command to display this field.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 8

----------- Criteria - Enter 1 or 2 to specify expression by field -----------
1 Id : ____________________________________________________________________ +
2 Sel: ____________________________________________________________________ +
Offset      0
Cmd Seq SHE Key Ref Field Name                        Picture  Type Start Length
                    ****  Top of data  ****
___ ___     ___   1 1 REC-TYPE01                                 AN      1     80
___ ___     ___   2  2 REC-TYPE                      XX         AN      1      2
___ ___     ___   3  2 NAME                          X(20)      AN      3     20
___ ___     ___   4  2 EMPLOYEE-NO                   9(4)       BI     23      2
___ ___     ___   5  2 AGE                           9(4)       BI     25      2
___ ___     ___   6  2 SALARY                        9(7)       PD     27      4
___ ___     ___   7  2 MONTH OCCURS 12 TIMES         9(8)       BI     31      4
___ ___     ___   8  2 FILLER                        XX         AN     79      2
                    ****  End of data  ****









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 79. Field Selection/Edit showing the Key field

2. In the Key fields adjacent to the fields you want to use in your multi-segment key, enter a number between one and sixteen.

   Your numbers do not need to start at one and do not need to be sequential, but must be unique. The numbers provided indicate the sequence of the fields within the multi-segment key. If you are using a keyed data set (for example, VSAM KSDS) and want to include the data set's intrinsic key in your multi-segment key, you must start the numbering at two or higher. (The intrinsic data set key is always included at position one in the key sequence.)
3. Save your template.

When your modified template is used in the compare process, the position and length information of the selected fields is pre-loaded into the appropriate sequence slots of the Keyed Synchronization Settings panel.

Comparison output

This section provides a general description of the comparison report produced by the File Manager Compare Utility, and lists the special flagging characters that can appear in the report to indicate various conditions. See the "Compare Utility: Options panel" on page 479 for a description of the different listing types and how the listing options affect the report.

If you run the comparison in the foreground and you have your print output directed to the File Manager print data set, the Print Browse function is invoked to display the comparison report.

Report sections

The File Manager compare utility report contains three sections:

Prologue
The prologue is generated unless you select report **Listing Type** = “none”. It lists the data sets being compared, and any copybooks or templates that were specified to filter or format the records.

Body The report body is generated unless you select report **Listing Type** = “summary” or **Listing Type** = “none” The contents of the body is determined by the **Listing Type** (delta, matching or long) and the **Listing Options**, and whether the comparison type is record or formatted.

Summary
The summary is generated unless you select report **Listing Type** = “none”. It tallies the number of records processed, the number of matching records found, the number of insertions and deletions found, the number of records that were not selected by record identification and selection criteria processing, and the number of records that were not compared due to invalid data in a formatted comparison.

The number of records not selected or not compared is only reported for non-zero totals. The number of records processed includes the "not compared" total but excludes the "not selected" total. A summary of the type of comparison and the options specified is provided. For keyed comparisons, the key segment positions and lengths are also listed.

Template details
Template details are reported when you select one of the template reporting options, unless you also select **Listing type** = “none”. For each template used in a formatted comparison, the template fields can be listed along with their attributes (**Show template layouts** option). Identification and selection criteria in the templates can be listed (**Show template criteria** option) for both record comparisons and formatted comparisons. For formatted comparisons, the template mapping information can also be listed (**Show mapped fields** and **Show unmapped fields** options). You can choose the manner in which template field information is reported by means of the **Show array elements**, **Show start/end as hex offset**, and **Show length in hex** options.

If you have selected the **Show template layouts** option, each 01-level is reported, followed by each of its fields. For each field, the field name, picture clause, type, start, end, and length are reported. If possible, each field is reported on a single line, and the report layout is adjusted to allow for the maximum length of the content of the field name and picture columns. If necessary, the field name, picture clause, and remaining information is reported on separate lines.

If you have selected the **Show template criteria** option, and the template was not dynamically created from a copybook, then the identification and selection criteria expressions, if any, are reported. If either expression is not present, a message is written to the report to indicate this. Within the expressions field, names are appended to the reference identifiers as an aid to readability.

If you have selected the **Show mapped fields** option or the **Show unmapped fields** option, each new 01-level is reported, followed by each of its fields in the same format as the layout information report. Each mapped field is followed by similarly formatted information for the corresponding field from the mapped old template layout. The mapped old

field information is preceded by an equals sign (=) in the **Ref** column. Similarly, each unmapped (new) field is preceded by an asterisk (*) in the **Ref** column. A horizontal rule indicates the end of the new fields, and is followed by any unmapped fields in the old template layout. These fields are also preceded by an asterisk (*) in the **Ref** column. Whether mapped or unmapped fields or both are present in the mapping information report is determined by which mapping reporting options you selected.

Body section record formats

Within the body section, any records that are printed are presented in one of four basic ways, depending on the type of comparison (record or formatted) and whether the “wide listing” option is selected. Variations on the basic layouts are created if the “show hex chars” or “highlight changes” options are selected.

Narrow record report

This report is produced when the “record” Comparison Option is selected but the “wide listing” Listing Option is not selected.

The width of the report is restricted to 132 characters, with long records being split over several lines. For matching records, only the “New” record is reported. For paired insertions and deletions and changed records, the “New” record is reported first, followed by the “Old” record. For unpaired insertions and deletions, the “orphaned” record is reported. When both the “New” and “Old” record must be split, the record sections are interspersed so that corresponding bytes remain vertically adjacent.

This format is useful if you need to print the comparison report.

Wide record report

This report is produced when both the “record” Comparison Option and the “wide listing” Listing Option are selected.

The report uses the full width of the File Manager list data set. Matching and non-matching records are reported or suppressed in the same way as for the narrow record report.

This format is generally easier to analyze than the narrow record report if you only want to review the report online.

Narrow formatted report

This report (also referred to as “single” format) is produced when the “formatted” Comparison Option is selected but the “wide listing” Listing Option is not selected.

The width of the report is restricted to 132 characters, split vertically to display fields from the “Old” and “New” data sets. Fields from the “New” data set are placed on the left side of the report, and fields from the “Old” data set are placed on the right side.

For matching records, only the “New” record fields are reported, but the “Old” record number is still reported on the right to help distinguish matching records from unpaired insertions. For paired insertions and deletions and changed records, fields from both records are reported. For unpaired insertions and deletions, the fields from the "orphaned" record are reported and the other half of the report is left blank.

This format is useful if you need to print the comparison report.

Wide formatted report

This report (also referred to as "table" format) is produced when both the “formatted” Comparison Option and the “wide listing” Listing Option are selected.

Like the wide record report, it uses the full width of the File Manager list data set, with each field formatted across the width of the report. The formatting of the fields is adjusted from normal print formatting so that the “Old” and “New” data can be vertically aligned as best as is possible for the mapped fields.

This format is generally easier to analyze than the narrow formatted report if you only want to review the report online.

Body section flags

The following flags can appear in the body section to indicate various conditions. For record comparisons and wide formatted comparisons, the flags appear in the ID column at the left of each line of data, unless otherwise indicated. For narrow formatted comparisons, the flags appear to the left of the record header information that precedes the fields for a record, or to the left of the field name.

I Indicates an inserted record in the “New” data set.

D Indicates a deleted record in the “Old” data set.

C Indicates a changed record in the “New” data set when the comparison was performed using keyed synchronization. This highlights the special case where the records had matching keys but altered data.

O Indicates the old data in a changed record when the comparison was performed using keyed synchronization.

+ Indicates a key sequence error in a “keyed” synchronization comparison.

- Indicates a key truncation error in a “keyed” synchronization comparison.

***** Indicates both a key sequence error and a key truncation error in a “keyed” synchronization comparison.

x Indicates invalid key data in a keyed synchronization comparison. This flag can only appear when one of the key segments has been defined with the packed data type, and the data in that segment in the current record is not valid packed decimal data.

! Indicates a record that could not be compared in a formatted comparison because one of the fields contained invalid data, or because a field was partly or wholly outside the record. It also indicates a record whose type is not included in the template mapping.

? Indicates a record that could not be compared in a formatted comparison because one of the fields in the corresponding record from the other data set contained invalid data, or a field that was partly or wholly outside the record. It also indicates a mapping mismatch between the record type identified for the new record and the record type identified for the old record. In this case, both the new and old records in an insert/delete pair are flagged with "?". Changes are not highlighted for the highlight changes option because there is no mapping to define which fields to compare.

| If the “highlight changes” Listing Option is selected with reports other than “narrow formatted”, this appears beneath the “Old” data set record in a paired insertion and deletion or changed record to indicate the changed data.

***** If the “highlight changes” Listing Option is selected for a “narrow formatted” report, this appears to the left of the new field name in a paired insertion and deletion or changed record to indicate the changed field.

When arrays are being compared and the number of array elements in the old record is different to the number of array elements in the new record, the array elements which are not matched are reported as changed fields.

Template details section

The template details section appears after the comparison summary. The information it contains depends on the combination of template reporting options you specify, the comparison type, and the information contained in the templates used in the comparison. Subject to these conditions, the overall order of presentation is as follows:

```
(page break)
old template report:
  first old layout
  first old layout criteria
  second old layout
  second old layout criteria

.
.
.
(page break)
new template report:
  first new layout
  first old layout criteria
  second new layout
  second new layout criteria

.
.
.
(page break)
mapping report:
  data for mapped layouts:
    first mapped new layout fields and associated old fields
    unmapped old fields associated with first mapped new layout
    second mapped new layout fields and associated old fields
    unmapped old fields associated with second new layout map

.
.
.
  data for unmapped layouts:
    first unmapped new layout
    second unmapped new layout

.
.
.
    first unmapped old layout
    second unmapped old layout

.
.
.
```

Note:

1. If the old and new templates are the same, allowing for the application of any offsets, then only one template report is produced. In this case, unmapped old fields are only reported when a layout is not mapped to itself, and unmapped old layouts are not reported at all (as they would just duplicate the unmapped new layouts).
2. Mapped old fields and unmapped old and new fields are flagged in the mapping report as described in the next section.

3. A separator line consisting of hyphens appears between the mapped new layout fields and associated old fields, and any unmapped old fields associated with a mapped new layout.
4. Any unmapped old layouts are separated from any preceding unmapped new layouts by the text "Unmapped old template layouts:".

Template details flags: The following flags can appear in the template details mapping report to indicate the relationship between old and new data set template fields. The flags appear at the left side of the **Rel** column.

= Indicates the flagged old data set template field is mapped to the preceding new data set template field.

* Indicates the flagged old or new data set template field is unmapped.

Formatted comparison of unlike fields

When a formatted comparison is requested and the template mapping specifies that fields with unlike attributes are to be compared, File Manager attempts to compare the data in the fields in a meaningful way.

The first step is to classify each of the fields as being numeric data, bit string data, or character data. All internal (binary, packed decimal and floating point) and external (zoned decimal and external floating point) numeric data types are classified as numeric. Bit fields up to 64-bits long are classified as bit strings. All other fields, including any grouped items that have been mapped, are classified as character data.

The next step is to determine how to compare the fields. The following table shows what comparison method is used for each combination of data classifications (redundant combinations have been left blank).

Table 4. Comparison method for formatted comparison of unlike fields

Data type	numeric	bit string	character
numeric	numeric	numeric	numeric
bit string		bit string	bit string
character			character

Numeric comparison
Fields that are not defined as internal numeric types are first converted to an internal numeric type compatible for comparison with the other field. If the conversion fails, the fields do not match. Packed and zoned decimal fields are validated before comparison, and if the validation fails, the fields do not match. Other than conversion and validation errors, and errors of precision for floating point numbers, the data types and lengths of the fields do not affect the result of the comparison.

Note: A character field containing an (external) floating point number is valid even if the field with which it is being compared is not defined as floating point.

Bit string comparison
Bit string comparison results in a match if corresponding bits in the two fields all match.

Character comparison
The result of a character comparison depends on which additional

formatted Comparison Options were specified. If no additional options were specified then the fields must be the same length and corresponding bytes in each field must match exactly. You can use the additional options to specify that either leading or trailing blanks, or both, are ignored when a character comparison is performed. You can also specify an option that causes fields to match even when blank-delimited "words" have been reformatted in the field. With this option, the leading and trailing blanks options are redundant, because any contiguous series of blanks in either field is treated as if it were a single blank. To compare character fields of different lengths, specify one of these options, otherwise none of the records match. Finally, you can specify an option to ignore case when comparing character fields.

Note: Records that fail validation are flagged with an exclamation mark (!) in the comparison report. For a wide (table format) listing the flag appears in the "ID" column. For a narrow (single format) listing the flag appears to the left of the record heading.

Comparing load modules

Load modules are, as members of libraries, sequential datasets, and can be compared by content like other datasets. A record-mode compare helps to check whether load modules are the same or not, but as the result contains differences between module texts, a further analysis is usually required.

However, in most cases, you are likely to want to see differences at a logical level, rather than at record level. You may want to know, for example, if both versions of a module were linked at the same time, and if CSECTs differ in size or were compiled using the same compiler versions.

A formatted comparison of load modules allows you to compare load modules at the logical level, with the ability to select properties to be compared and reported

Conceptually, you can see the process of a formatted comparison of two load modules in the following way:

- You define information you want to compare, by selecting compare criteria at the load module level:
  - Load module size.
  - Load module entry point address.
  - Version of the linkage editor or binder used to prepare the load module.
  - Load module link date and time.
  - AMODE and RMODE of load module.
  - Load module authorization code (AC).
  - Load module link attributes.

  And at the CSECT level (if you want CSECTs to be compared):
  - CSECT size.
  - CSECT address.
  - Version of a language compiler used to compile the CSECT.
  - CSECT compile date.
  - AMODE and RMODE of CSECT.
  - AMASPZAP IDR data.
  - Text - CSECT content (instructions and constants).
- For each of the compared load modules, File Manager creates one or more logical records containing information extracted from load module. Load module properties, such as load module size, Entry Point Address, and load module link date, constitute fields of one logical record. Each CSECT has its own dedicated

logical record containing CSECT properties, such as CSECT size, address, and compilation date. If additional CSECT information, ZAP IDR or CSECT text, need to be compared, the logical record is split into a number of physical records, with the layout appropriate to the information they hold.

- There is a natural mapping determined by the layouts of the records and the compare criteria you selected. It defines the compare process as a formatted comparison. The record representing information at the load module level is compared with the corresponding record representing the second module. Corresponding records (records of the same CSECT name) representing CSECTs are also compared. The comparison report shows differences in a similar way to any other type of formatted comparison. Indentation shows the hierarchy of information: load module ► CSECT ► CSECT text and ZAP IDR data.

You can perform load module comparison using either a Compare Utility (3.11, or DSM in batch), or its dedicated sub-function, Load Module Compare Utility (3.10.2, or CLM in batch). They share the same ISPF variables and, in batch, have the same syntax.

To perform a formatted compare of load modules:

1. From the Primary Options menu, select Compare Utility (option 3.11).

   File Manager displays the Compare Utility - "Old" Data Set Entry panel.
2. Specify the "Old" load module information.
3. Press Enter.

   File Manager displays the Compare Utility - "New" Data Set Entry panel.
4. Specify the "New" load module information.
5. Press Enter.

   File Manager identifies the load module on both "Old" and "New" sides and assumes that you want to perform a load module formatted comparison. Accordingly, File Manager displays the Compare Load Module Options panel, instead of the Compare Options panel.
6. Specify:
   - **Comparison type**: "Formatted", unless you want the content of the load modules to be compared in record mode.
   - **Compare level**: "Module" level is more general than the "CSECT" level.
   - **Load module criteria**: specify the load module properties you want to compare. If no criteria is selected, then only a comparison of the load module name is performed.
   - **CSECT criteria**: specify the CSECT properties you want to compare (only applicable if **Compare level** is CSECT).
   - **Processing Options**: specify as required.
   - **Listing Options**: specify as required.
7. Press Enter to perform the compare.
8. When you have finished viewing the report or editing the batch JCL, press the Exit function key (F3) to return to the "Old" data panel.

Related topics

“Compare Utility: “Old” and “New” panels” on page 471
“Compare Utility: Load module options panel” on page 475

Printing from File Manager

Using File Manager, you can print entire data sets, selected records from data sets, audit trail reports, template definitions and various other reports generated by File Manager utilities.

Your print output can be directed to the current SYSPRINT allocation, a temporary data set, a REXX stem variable, or to your terminal, depending upon your Print Processing options. You can also use these options to specify the format and layout of your print output.

If you are working in online mode and your print output has been directed to a temporary data set, you can send that data set to a printer by taking the following steps:

1. From the Command line of any menu or Entry panel, enter the PB command (Print Browse), or for the File Manager debug trace, enter the PBT command (Print Browse Trace). This displays the current contents of your temporary print or trace data set.
2. Enter the PRINT command. This sends the data set to the SYSOUT class *c* allocated to a printer in your work environment.

Setting your Print Processing Options

The Print Processing Options control such things as the destination of your print output, the page format to use and the number of records to print.

Additional Print formatting options information (redefined fields, field reference number, field type and length values, picture clause, start location, structure) are shared with the Edit/Browse options and can be set using Edit/Browse options (0.6) in Settings.

To set your options:

1. From the File Manager Primary Options Menu, select option **0 Settings**. The Set Processing Options Menu panel is displayed.
2. Select option **1 Print**. The Set Print Processing Options panel is displayed.
3. Set your print processing options:

Output destination
Where you want the printed output sent:

SYSPRINT
Send print output to the current SYSPRINT allocation.

Typically, SYSPRINT is allocated to the terminal, making this option synonymous with TERMINAL. However, you can allocate SYSPRINT in other ways.

Terminal
Send print output to the terminal.

Data set
Output is accumulated in the print data set specified in the **Data set name** field. This data set can browsed using the PB command or sent to the JES spool queue for printing by issuing the PRINT command while browsing the data set. (The output is sent to the JES spool queue class specified in the Output class field.)

REXX Send print output to the REXX stem variable FILEM.

Choose this option when you want to control your printing from within a REXX program.

Record length
How many columns wide the output is

Page skip
Whether output from each function starts on a new page

Wide print
Whether the maximum print line length for print output is used

Translate non-printable chars
Whether non-printable characters are translated to blanks

Uppercase message text
Whether all message text is translated to uppercase.

Data prefix
Whether the output contains a data header for each record

Header page
Whether the output contains a header page at the beginning

Data set DISP
Whether the print output is appended to the existing data set (MOD) or replaces it (OLD).

Note: This option only affects output sent to the print data set specified in the PRINTDSN option.

Dump format
Which format (updown or across) is used for hexadecimal print output

Data set name
The print data set where print output is directed when the PRINTOUT print option is set to SYSOUT=*c*

Output class
The class of the JES spool queue to be used when the PRINT command is issued while browsing the temporary print data set.

Lines per page
How many lines per page the output has

Record limits
Which part of each record to print

4. Press the Exit function key (F3) to save your changes and exit from the panel.

Related topics
"Set Print Processing Options panel (option 0.1)" on page 658
"Editor Options panel" on page 546

Printing data sets

You can use the Print Utility to print supported data sets or members in a selected format. Concatenated like and unlike sequential data sets are supported. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for print processing. Such invocation may fail as DFSORT does not allow for unlike concatenation of data

sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

You can print data by record, by block (non-VSAM) or control interval (VSAM), or by field using a template. You can select records for printing using any (or none) of these:
- The start key (VSAM only).
- Skip and print count fields.
- Criteria and record type selection defined in a template.

You can use a template to:
- Select the records you want to print
- Select the fields you want to print
- Specify the number of columns allocated to each field when printing in TABL format
- Suppress leading zeros for numeric fields
- Modify field headings

In SNGL print format, additional record information (redefined fields, field reference number, field type and length values, picture clause, start location, structure) can be printed depending on the Edit/Browse options. Numeric fields can be left-justified depending on the options.

An example showing how the print output in SNGL print format can vary (depending on the Edit/Browse options) is shown in "Printing a single record" on page 306.

You can also use a DFSORT or REXX procedure, contained in a specified PDS member, to further enhance the Print Utility.

The Print Utility can be run in foreground or as a batch job.

To print a data set:
1. From the Primary Option Menu panel, select **3. Utilities**.
2. From the Utilities Menu panel, select **2. Print**. The Print Utility panel shown in Figure 211 on page 621 is displayed.
3. Perform either of these actions
   - Specify the data set or PDS member that contains the data you want to print, using a combination of the **Data set name**, **Member**, and **Volume serial** entry fields
   - To select a range of members based on selection criteria, specify the data set name or name pattern for a PDS(E) data set, optionally the member name or pattern, and select the **Advanced member selection** option

   Note: If you enter details in the **Member** field and select the **Advanced member selection** option, File Manager populates the **Member name** field on the subsequent Advanced Member Selection panel with the same value.
4. Select the data that you want to print, at either the record level or field level:
   - For record-level selection, use the following entry fields on the Print Utility panel:

     Start key
     The starting key or slot location for a VSAM file

Skip count
The number of records to be skipped before processing begins

Include
The number of physical records that to be included in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Repeat skip
The number of physical records to be skipped in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Print count
The number of records to be printed

- For field-level selection, use a template.

If you specify both record-level and field-level selection, File Manager first selects data at record level, then at field level.

5. Specify your processing options to control the way in which the data is selected and printed.

   You can specify:
   - The print format
   - If a template is to be used
   - Block or control interval printing
   - If you want to edit a template
   - Batch or foreground processing
   - A procedure to be called by the Print Utility

6. If your data is in a sequential data set or PDS or PDSE member, is in ISPF PACK format and you have not specified an I/O exit, you can select an ISPF Packing option.

   1. Unpack
   Select this option when you want File Manager to unpack the data before any user processing or printing occurs.

   2. None
   Select this option when you do not want File Manager to detect the ISPF PACK format and to operate on any packed data in its packed format.

   This is the only valid option when your data is not a sequential or PDS data set, or when the Use I/O exit option has been used.

   3. Skip
   Select this option when you want File Manager to detect data in ISPF PACK format and stop any processing or printing for the data set or member.

7. Additional print formatting options are available after selecting the **Additional print options** option. These options allow for the printed output to contain additional information about record structure, as well as controlling the inclusion of records when their length does not match the layouts of the template records.

8. Press Enter.

   If you selected the **Advanced member selection** option, File Manager displays the Advanced Member Selection panel:

   a. Specify the selection criteria to select the members you want to print and press Enter.

b. If you have left the **Member name** field on the Advanced Member Selection panel blank or have entered an asterisk (*) or a mask, File Manager displays the Member selection panel. In this case, select the members you want to print (either by typing an "S" in the **Cmd** field, or with the SELECT primary command) and press Enter.

File Manager sends the output to the location specified in your Set Print Processing Options.

If you have sent the output to a print data set, you can view the data set using Print Browse (option 3.9). While you are using Print Browse, you can use the PURGE primary command to clear the print data set. To transfer the contents of the print data set to a SYSOUT class, press the Print function key (F4).

By using a print data set in this way, you can avoid print width truncation that might occur with TABL format printing.

Related topics
"Print Utility panel" on page 620
"Managing templates" on page 140
"Setting your Print Processing Options" on page 299
"Selecting a range of PDS(E) members" on page 34
"Advanced Member Selection panel" on page 439
"Editor Options panel" on page 546
"Print Options panel" on page 619

Using a DFSORT or REXX procedure

You can specify an existing DFSORT or REXX procedure, or create a new one, to further enhance the way in which data is selected and printed.

If you have a template, first use the selection criteria within the template to select your data. For more sophisticated tasks, use DFSORT statements to select data, manipulate the output records, and to write to other files. Use REXX to execute other functions that are provided by File Manager, and to perform complex tasks that cannot be achieved any other way.

Note: In REXX procedures for the Print Utility, avoid using the PRINT() function, as its output is interspersed with the output from the Print Utility.

For general information about using DFSORT or REXX procedures, see Chapter 13, "Enhancing File Manager processing," on page 405.

Example 1

This is a DFSORT procedure that prints all records where the surname at column 10 is "Smith".

```
*FASTPROC
 OMIT COND=(10,10,CH,NE,C'SMITH ')
```

Example 2

This example uses REXX to report on total salaries paid, where the salary is a packed number starting at column 20 for a length of 7.

```
  rc = tally(20,7,P,'Total Salaries Paid')
```

REXX member selection

For PDS or PDSE data sets, you can determine whether a member should be printed or not printed, depending upon a condition being true within the member

records. This conditional test is supplied to File Manager via a REXX procedure that includes at least one of the following RETURN strings:

RETURN PROCESS MEMBER
Indicates that the member is to be included in the print. No more records in this member are passed to the REXX proc. The member is printed intact, subject to any specified template processing, which is performed before the user REXX proc is invoked.

RETURN DROP MEMBER
Indicates that the member is to be excluded from the print. No more records in this member are processed. Processing continues with the next member.

When REXX member selection is in effect, records are read from an input member and then cached in memory until the decision is made whether the member is to be printed. Once the decision has been made, the entire member is either printed or dropped, depending upon the return string specified.

If the entire member is processed without encountering a DROP MEMBER or PROCESS MEMBER return string, the member is processed according to the specified default action. If either of these strings is returned by the REXX processing when REXX member selection HAS NOT been specified, it is treated as if it was a RETURN with no argument strings, except that a warning message is issued. Subsequent records continue to be passed to the REXX proc. Similarly, if a DROP, STOP or STOP IMMEDIATE string is returned by the REXX processing when REXX member selection HAS been specified, it is treated as if it was a RETURN with no argument strings, except that a warning message is issued. Subsequent records continue to be passed to the REXX proc until a decision has been made on whether to DROP or PROCESS the member.

Other REXX statements that control the dropping or updating of records are not performed when REXX member selection is used. Any changes to the special File Manager-defined REXX variable OUTREC are ignored. However, REXX statements that do not involve updating or dropping records are performed. This may lead to some unexpected results. In general, if you need to print members of a PDS conditionally based on the contents of the member, and also want to manipulate the contents of the member, try to execute this task as a two-step process using separate REXX procedures.

To specify conditional member processing:

1. On the Print Utility panel, specify your data set name and other print options.
2. Select the REXX member selection option and specify the default action (P or D) in the adjacent field. If this field is left blank, P is assumed. P denotes process or print the member, D specifies drop or do not print the member.
3. Select the Use proc option and specify an existing REXX procedure or enter an * in the field to create a new REXX procedure. This procedure must contain a conditional test and at least one of the relevant RETURN strings.
4. Complete the printing process as required.

Example 1

In this example, the SUBSTR function is used to test records in each member being processed. When the procedure encounters a record in which the condition tests as True, the entire member is dropped from processing. All other members are printed or not printed, according to the default setting on the Print Utility panel.

```
if substr(inrec,4,3) = '333'
then RETURN DROP MEMBER
return
```

Example 2

This example demonstrates what happens when you use REXX member selection at the same time as other REXX procedure statements. In this scenario, you are attempting to print members in a data set and to tally a value in those members at the same time.

The first member (MEM1) contains 5 records:

```
AAA111BBB456CCC789
AAB222BBB456CCC789
AAC333BBB456CCC789
AAA444BBB456CCC789
AAA555BBB456CCC789
```

The second member (MEM2) contains 4 records:

```
AAA001BBB456CCC789
AAA002BBB456CCC789
AAA003BBB456CCC789
AAA004BBB456CCC789
```

The Print Utility REXX member selection default is set to 'P', and the following procedure is supplied.

```
/* rexx */
if substr(inrec,1,3) = 'AAA'  /* If cols 1-3 in current rec contain 'AAA' */
then x = tally(4,3,Z,'total')   /* then add up the value in columns 4-6 */
if substr(inrec,4,3) = '333'  /* If cols 4-6 contain '333' */
then RETURN DROP MEMBER   /* then drop member from print processing */
if substr(inrec,4,3) = '003'  /* If cols 4-6 contain '003' */
then RETURN PROCESS MEMBER  /* then process (print) member */
return
```

The tally for “total” includes the first record from MEM1, even though MEM1 is dropped from the print process. The second and third records don't satisfy the 'AAA' requirement and so aren't tallied, and the subsequent records won't be processed because the decision to drop has been made while processing the third record.)

The tally for “total” also includes the first three records in MEM2. The fourth record is not processed by the REXX proc because the decision to print has been made while processing the third record. The end result of this print action is that MEM1 is dropped, MEM2 is printed, and the tally for “total” equals 117. (111 + 1 + 2 + 3).

Printing DBCS data

When printing in character or hexadecimal format, you can also print double-byte character set (DBCS) data, if you have the necessary hardware. Use Set DBCS Format (option 3.0) to specify which data in each record is in DBCS format, mixed format, or EBCDIC format within an input record; and use Compiler Language Selection (option 0.5) to specify a DBCS-capable language.

If you want to align DBCS and EBCDIC data in printed output, use a font that sets the width of each DBCS character to 2 EBCDIC bytes.

If printing in TABL or SNGL format, the format definition is taken from the template, and any DBCS formatting set by this option is ignored.

When you select the Set DBCS Format option, the Set DBCS Format panel shown in Figure 227 on page 652 is displayed.

You can define up to 32 fields in ascending order. They must not overlap, but they need not be contiguous. Specify for each field:

Start column
The first column in the record where the data is located.

End column
The last column in the record where the data is located. An asterisk (*) indicates end of record.

Type The type of data:
EB EBCDIC
DB DBCS

These are graphics fields, that is, data that is DBCS, but stored without the shift-out and shift-in characters.
MI Mixed format

For all areas you do not define, File Manager creates an EBCDIC field definition.

Note:

1. To cause the type DB to correctly format DBCS characters for printing, the language used for File Manager must be a DBCS-capable language.
2. To print a listing of the current format definitions, enter the LIST primary command.
3. To restore your definitions to the default value of `1,*,EB`, enter the RESET primary command.

Printing a single record

At times, you may want to print a single record. Instead of setting up your template and selection criteria to narrow your selected records down to the one you require, you can print individual records. To do this:

1. Display your data in a Browse or Edit panel.
2. Select your display format (SNGL, TABL, CHAR, HEX or LHEX).
3. Scroll your data until the record that you want to print is at the top of the data area.
4. Enter one of the following primary commands:
   - RD (“Record Dump”) - this prints the current record in dump format, with hexadecimal values under the record data.
   - RP ("Record Print") - this prints the current record. The format of the output from the RP command depends on the display format when you enter the RP command. While printing records in SNGL view, additional information (redefined fields, field reference number, field type and length values, picture clause, start location, structure) can be printed depending on the Edit/Browse options. Numeric fields can be left-justified depending on the options.

Examples

Here is an example of RD output when the display format is CHAR:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
                      CHAR CToyota             Avalon             AVALONConq
                      ZONE CE9A9A844444444444444CA899944444444444444CECDDDC999
```

```
NUMR 3368631000000000000001513650000000000000001513653658
    01---+----1----+----2----+----3----+----4----+----5-
CHAR   Duncan Autos             SedanAuto62995PSteelYYYY
ZONE 29CA98894CAA9A4444444444444E8889CAA9FFFFFDEA889EEEE
NUMR 8C44531501436200000000000002541514366299572355388888
   101---+----1----+----2----+----3----+----4----+----5-
```

Here is an example of RP output when the display format is CHAR:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
IBM File Manager for z/OS
                            CToyota              Avalon              AVALONConq
                              Duncan Autos             SedanAuto62995PSteelYYYY
```

Here is an example of RP output when the display format is SNGL (with all additional formatting options selected):

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
Ref Field                      Picture Typ Start   Len   Data
-------------------------------------------------------------------------------
Record Number - 8

  1 1 VEHICLE-REC                      AN      1   159
  2  2 VEHICLE-TYPE             X       AN      1     1   C
  3  2 VEHICLE-MAKE             X(20)   AN      2    20   Toyota
  4  2 VEHICLE-MODEL            X(20)   AN     22    20   Avalon
  5  2 MODEL-CODE               X(6)    AN     42     6   AVALON
  6  2 VEHICLE-SUB-MODEL        X(20)   AN     48    20   Conquest
  7  2 YEAR-OF-MANUFACTURE      9(4)    ZD     68     4   2000
  8  2 NUMBER-ADVERTISED        9(4)    ZD     72     4   1
  9  2 FOR-SALE OCCURS 1 TO 100 TIMES DEPENDING ON NUMBER-ADVERTISED
     2 FOR-SALE(1)                     AN     76    84
 10   3 REGO-NUMBER             X(8)    AN     76     8   1ARW-832
 11   3 COLOUR                  X(8)    AN     84     8   Green
 12   3 ODOMETER                X(7)    AN     92     7   53864
 13   3 ASKING-PRICE            S9(6)   PD     99     4    25289
 14   3 DEALER                  X(25)   AN    103    25   Duncan Autos
 15   3 BODY-TYPE               X(5)    AN    128     5   Sedan
 16   3 TRANS-TYPE              X(4)    AN    133     4   Auto
 17   3 ENGINE-TYPE             9       ZD    137     1   6
 18   3 ENGINE-CAPACITY         9(4)    ZD    138     4   2995
```

Here is an example of RP output when the display format is SNGL (with no additional formatting options selected):

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
Field                     Data
-------------------------------------------------
Record Number - 8

VEHICLE-TYPE              C
VEHICLE-MAKE              Toyota
VEHICLE-MODEL             Avalon
MODEL-CODE                AVALON
VEHICLE-SUB-MODEL         Conquest
YEAR-OF-MANUFACTURE       2000
NUMBER-ADVERTISED         1
REGO-NUMBER(1)            1ARW-832
COLOUR(1)                 Green
ODOMETER(1)               53864
ASKING-PRICE(1)            25289
DEALER(1)                 Duncan Autos
BODY-TYPE(1)              Sedan
TRANS-TYPE(1)             Auto
ENGINE-TYPE(1)            6
ENGINE-CAPACITY(1)        2995
```

Here is an example of RP output when the display format is TABL:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
VEHICLE-TYPE VEHICLE-MAKE         VEHICLE-MODEL        MODEL-CODE VEHICLE-SUB-MO
AN 1:1       AN 2:20              AN 22:20             AN 42:6    AN 48:20
-            <---+----1----+----> <---+----1----+----> <---+>     <---+----1----
C            Toyota               Avalon               AVALON     Conquest
```

Related topics

"Browse panel" on page 455
"RD primary command" on page 801
"RP primary command" on page 811

Printing your Audit Trail Report

Use the Print Audit Trail Report function to print an audit trail report using the contents of the audit trail data set. An audit trail data set is created if:

- The **Create an audit trail** option is selected when you use either the Edit data or Edit MQ data function.
- The auditing setting in the FMN0POPT macro has been set to YES or DEMAND.
- A SAF rule is in place for the function and resource.

Note: You can create an audit trail data set yourself using the batch job FMNSMFX (supplied with File Manager). This job creates an audit trail data set using data from the SMF log file. For details, see the *File Manager Customization Guide*, SC19-4118.

When you select the **Print audit trail report** option, the Print Audit Trail panel shown in Figure 209 on page 618 is displayed.

To print an audit trail report:

1. Specify the name of the audit trail data set that contains the audit trail data you want to print (**Data set name**).

   If File Manager created the audit trail data set, the name is in the format *userid*.FMNLOG.D*yymmdd*.T*hhmmss* where:

   userid is the ID of the user

   FMNLOG
   is a constant that identifies the data set as an audit log (trail)

   D*yymmdd*
   is the date File Manager created the audit trail data set

   T*hhmmss*
   is the time File Manager created the audit trail data set

   Note: If the AUDITHLQ option is set in the FMN0POPT macro (see *File Manager Customization Guide*), audit trail data sets have the name *audithlq.userid*.FMNLOG.D*yymmdd*.T*hhmmss*.

   If you created the audit trail data set yourself, use the data set name you specified in the supplied batch job FMNSMFX. For details, see the *File Manager Customization Guide*.
2. To help identify your audit trail report, use the **Description** entry field. If the description contains imbedded spaces, you must enclose it in quotes. The description you enter appears on the Formatted Audit Event Records section of the report.
3. Select the processing options you want:

4. Press Enter.

Audit trail report

The audit trail report is divided into two sections: Detailed Statistics and Summary Statistics.

The Detailed Statistics section shows all audited inserted, deleted, read, written, and before and after change records. The format of this section of the report can be further controlled by the options available on the Print Audit Trail panel. If a template was used when accessing the data, the report is formatted.

The Summary Statistics section lists the totals of the records reported together with the function and resource information pertaining to this report.

Chapter 7. Using File Manager utilities

File Manager offers many utilities that you can use to work with your data, memory and storage systems.

This chapter provides information about and instructions for using these utilities. The major tasks and concepts are:

Managing catalog entries

To manage your catalog entries from within File Manager, you can use the Catalog Services Utility. Catalog Services uses IDCAMS system services to obtain and process catalog information. For instance, the effect of using the Alter, Define, and Delete services, is the same as using the equivalent IDCAMS command with the same parameters.

Note: To use Catalog Services effectively, you need a basic understanding of the catalog system and IDCAMS usage. For more information about IDCAMS, see *z/OS DFSMS Access Method Services for Catalogs*.

The tasks and concepts described in this section are:

Available catalog services

The services you can choose are:

List catalog entries
Displays a subset of catalog entries based on a generic name or entry type.

Display entry information
Displays the details of a single catalog entry.

Note: In batch, you can use the DSI function to print the details about a data set.

Print catalog entries
Prints a list of catalog entries that match the criteria (data set name, catalog name, and type) that you have specified. The printed output is produced according to the print options you have specified.

Alter catalog entry
Alters the definition of a single catalog entry.

Define catalog entry
Defines a new catalog entry, of the type specified. For non-VSAM entries, you can choose whether or not to allocate the data set as you are defining it. For other data types, the data set allocation is automatically done.

Delete catalog entry
Deletes a single catalog entry.

Related topics
"Printing from File Manager" on page 299

Selecting your service

To select an IDCAMS service:

1. From the Utility Functions menu, select **4 Dslist**. The Catalog Services panel is displayed.
2. Enter the **Data set name**:
   - Specify either a fully-qualified data set name or a generic data set name when you want to:
     – list catalog entries; or
     – print catalog entries.
   - Specify a fully-qualified data set name when you want to:
     – alter a catalog entry;
     – define a catalog entry;
     – delete a catalog entry; or
     – display entry information.
3. Optionally enter a catalog name. If you leave the **Catalog ID** entry field blank, File Manager uses the system catalog search order.
4. Select the **Entry type**.

   You can use the entry type processing option to restrict the catalog search to a specified type, or to search all types by selecting **Any**. The effect of this option depends on the catalog service you have chosen:

   - For all services, except "Define catalog entry", it simply restricts the search to entries of the specified type.
   - When you select an entry type (other than **Any**) with the "Define catalog entry" service, it predetermines or restricts the type of entry to be defined.

   Restricting the search (by using an entry type other than **Any**) is most useful with a generic data set specification.

   The entry type processing options are:

   Any[4a] Any catalog entry.

   Non-VSAM
   A non-VSAM data set entry. This type includes all non-VSAM data sets (such as PDSs, PDSEs, IAM and ordinary sequential data sets) but does not include non-data set types (such as aliases and GDGs).

   VSAM[4a]
   Any VSAM data set entry. This type includes all VSAM data sets, not just clusters and their components.

   AIX A VSAM alternate index entry.

   Alias An alias entry.

Cluster[4a 4b]
A VSAM cluster entry.

Data[4a 4b]
A VSAM data component entry.

GDG A generation data group entry.

Index[4a 4b]
A VSAM index component entry.

OAM[4c]
An Object Access Method entry.

Page space[4c]
A system page data set entry.

Path A VSAM path entry.

User catalog[4c]
A user catalog entry.

Note:

a. If you specify this entry type, and define a new catalog entry without using an existing catalog entry as a model, File Manager displays a pop-up menu, from which you can select a specific entry type.
b. When a component of a VSAM data set matches a search, all the components of the data set appear in the resulting list, whether they match the search or not. For example, if you have a VSAM ESDS with a cluster called FMN.ESDS and a data component called FMN.ESDS.D and specified the generic data set name `FMN.ES*` with entry type Data, both the cluster and data components are displayed, even though the type of the cluster is not a match. In fact, it would be displayed even if its entry name did not match.
c. These services are not supported:
   - "Define catalog entry"
   - "Delete catalog entry"
   - "Alter catalog entry"

5. To generate the data set list with all data sets matching the qualifiers in the **Data set name** field, including data sets with additional qualifiers, select the **Include Additional Qualifiers** option.
6. If you are defining, deleting, listing, or printing a catalog entry and you want to run the command in batch, select the **Batch execution** option.
7. To show all dates on the Data Set List panel and listing as YY/MM/DD (instead of the default YYYY.DDD), select the **YY/MM/DD date format (def. YYYY.DDD)** option.
8. To limit the number of data sets selected for VTOC or catalog processing, select the **Processing limit** option and type the maximum number of entries in the adjacent entry field.
9. Select one of the services listed at the top of the Catalog Services panel by typing the appropriate entry on the Command line. Leave the Command line blank for the "List catalog entries" service.
10. Press Enter. File Manager searches the system catalogs for the data set or sets you have specified and performs the service you have requested.

Note: Catalog Services attempts to help you to invoke IDCAMS such that your request is successful, but this is not always possible. If an error occurs, a pop-up window might be displayed containing IDCAMS messages.

Related topics
"Specifying a data set and a member name" on page 16
"Catalog Services panel" on page 464

Working with a list of catalog entries

When you select the "List catalog entries" option on the Catalog Services panel (by leaving the Command line blank), a list of catalog entries is displayed as shown in Figure 131 on page 466.

The list includes the following information for each entry:
- Name.
- Entry type.
- Primary volume.
- Multiple volume usage.
- Creation date (where available).
- The organisation of the data set.
- The record format of the data set.
- The logical record length of the data set.
- Number of extents.
- Number of kilobytes currently allocated for a VSAM, NVSAM or PDSE data set.
- The percentage of currently allocated space used for a VSAM, NVSAM or PDSE data set.
- Number of kilobytes currently available for a VSAM, NVSAM or PDSE data set.
- The percentage of currently available space for a VSAM, NVSAM or PDSE data set.
- The highest used byte as a percentage for a data set.
- The number of kilobytes available from the highest used byte to the current end of a data set.
- The percentage of space between the highest used byte to the current end of a data set.

To scroll to see more information, press the Right function key (F11).

Note: For an explanation of each item displayed on the Data Set List, refer to the online help.

If a catalog error occurs (for example, if the data component of a VSAM cluster is not found), ERROR appears in the **Entry type** field.

Note: You might be able to find out more about such errors by viewing the IDCAMS messages. To view these message, enter the AMSMSG command.

To update your list view, to include any changes that you have made and also any changes made by other users since you first displayed the list, enter the REFRESH primary command.

To view or find a data set entry, choose from the following techniques:
- To move forward or backward through the list, use the Down function key (F8) or the Up function key (F7). You can also enter a value in the **Scroll** field. You can also use the scroll commands to move the display window over the list.

- Use the FIND command to search for a character string that forms part or all of the data set name.
- Use the LOCATE command to scroll to a particular row number.

To select a service, perform either of these actions

- Type the line command:
  1. Move the cursor to the beginning of the line that contains the name of the entry that you want to work with.
  2. Type the command (for example, BROWSE). You can type an equal sign (=) to perform the same command for more than one entry.

  You can abbreviate the line commands to their first letter, (for example, A instead of ALTER) except in the case of EDIT and EXTENTS, where you would use the first two letters.
- Select the function from the Process pull-down menu:
  1. Move the cursor to the entry with which you want to work.
  2. Press the Process function key (F6). (If the cursor is outside the list area, you work with the first entry in the list.) The selected line is highlighted in another color. The Process pull-down menu opens below the action bar as shown in Figure 80.
  3. Select an option on the menu to perform a function such as browse or edit.

To define or delete a catalog entry and run the function in batch:

1. Select the **Batch execution** option on the Catalog Services panel.
2. Select the **with list** option on the Catalog Services panel.
3. On the Data Set List panel. enter the DEFINE or DELETE line command against the catalog entry you want to work with.

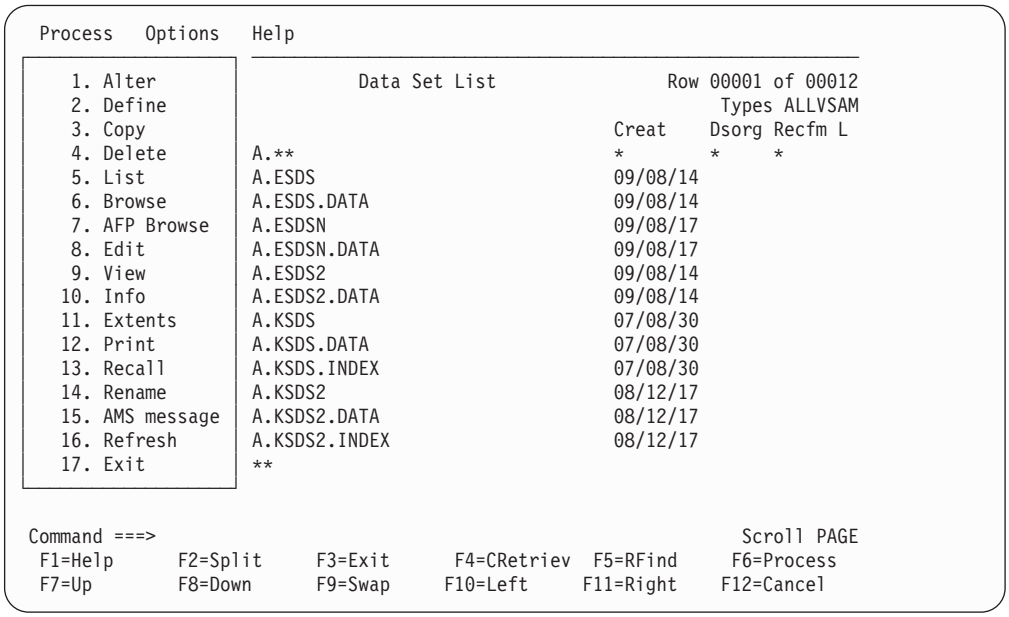


Figure 80. Catalog Services: list of catalog entries with Process pull-down menu

Related topics

"Scrolling to see data" on page 77
"Catalog Services Data Set List panel" on page 466
"FIND/FX primary command" on page 769

"LOCATE primary command" on page 786
"REFRESH primary command" on page 804
"Catalog Services Data Set List panel" on page 466
"Manipulating your view of selection lists" on page 28

Viewing your Catalog Entry Details

To view the Catalog Entry Details for your data set:

1. Perform either of these actions:
   - In the Catalog Services panel, type a fully qualified data set name (and optionally the catalog name) for the *existing* entry, then enter `I` on the Command line.
   - From the Catalog Services panel, display a list of data set names.
2. In the Data Set List panel, enter `I` in the line command field adjacent to the entry you want to view.

   An Entry Details panel for the entry type of your data set is displayed, showing the catalog information. For example, if your data set is a VSAM KSDS file, the VSAM Entry Details panel is displayed.

The Entry Details panel shows basic information about your data set. From this panel, you can display another panel that shows information about the associations for your data set. For VSAM files only, you can display another panel that shows the VSAM Extent information.

To display or alter any other information, you must use the IDCAMS utility. For more information about IDCAMS parameters, see *z/OS DFSMS Access Method Services for Catalogs*.

Related topics

- "AIX Entry Detail panel" on page 447
- "GDG Entry Detail panel" on page 581
- "IAM KSDS Define panel" on page 582
- "Non-VSAM Entry Detail panel" on page 609
- "Path Entry Detail panel" on page 612
- "VSAM Entry Detail panel" on page 722

Viewing association information

When there is more than one AIX, path or alias association for your entry, the Entry Details panel shows a message in the relevant field, for example:

```
VSAM Associations:
   Path . . . . . *** multiple paths defined over cluster ***
   AIX  . . . . . *** multiple AIXs associated with cluster ***
```

You can display these associations in a separate panel. To do this, enter the ASSOCS command, or press the ASSOCS function key (F6). The Associations Information panel appropriate for your current entry type is displayed.

To return to the Entry Detail panel from the Associations Information panel, enter the INFO primary command, or press the INFO function key (F6).

Note: If you do not alter the PF key assignments, F6 acts as a toggle switch between these two panels.

Related topics

- "AIX Association Information panel" on page 446
- "Non-VSAM Association Information panel" on page 607
- "VSAM Association Information panel" on page 719
- "ASSOCS primary command" on page 741
- "INFO primary command" on page 782

Statistics for VSAM data sets

To display more information about the statistics of a VSAM data set, enter the STATS command, or press the STATS function key (F11). The VSAM Extent Detail panel is displayed, showing VSAM statistics for every extent on every volume for the selected data set.

To return to the Entry Detail panel from the VSAM Extent Details panel, enter the INFO primary command, or press the INFO function key (F11).

Note: If you do not alter the PF key assignments, F11 acts as a toggle switch between these two panels.

Related topics

- "AIX Association Information panel" on page 446
- "Non-VSAM Association Information panel" on page 607
- "VSAM Association Information panel" on page 719
- "STATS primary command" on page 822
- "INFO primary command" on page 782

Displaying volume information

To display volume information for a VSAM or non-VSAM data set, from the VSAM Entry Detail panel or Non-VSAM Entry Detail panel, enter the VOLUME primary command, or press the Volumes function key (F5). File Manager displays a pop-up Volume Information panel (see Figure 81 on page 318) showing a list of allocated volumes for the selected data set.

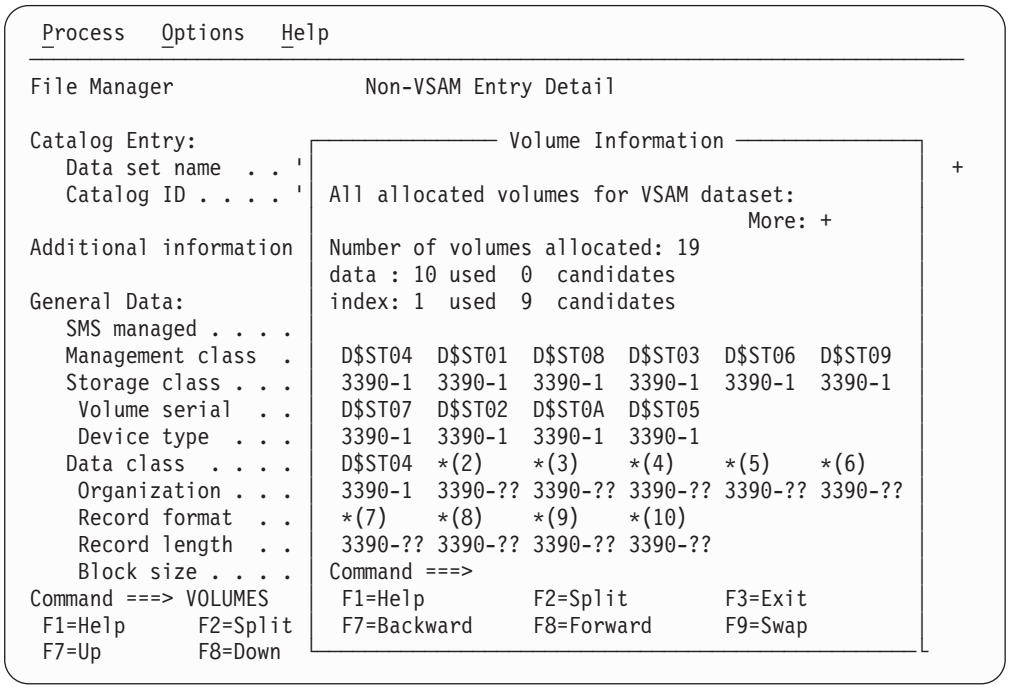


Figure 81.

To return to the VSAM Entry Detail panel or Non-VSAM Entry Detail panel from the pop-up Volume Information panel, press the Exit function key (F3).

Related topics
- "VSAM Entry Detail panel" on page 722
- "Non-VSAM Entry Detail panel" on page 609
- "VOLUME primary command" on page 829

Displaying extent details for non-VSAM data sets

To display extent details for a non-VSAM data set, from the Non-VSAM Entry Detail panel, enter the EXTENT primary command, or press the Extents function key (F11). File Manager displays the Non-VSAM Extent Information panel showing.

To return to the Non-VSAM Entry Detail panel from the Non-VSAM Extent Information panel, press the Exit function key (F3).

Related topics
- "Non-VSAM Entry Detail panel" on page 609
- "Non-VSAM Extent Information panel" on page 610
- "EXTENT primary command" on page 767

IAM Entry Detail display

IAM is a product that provides an alternative for VSAM (KSDS and ESDS) cluster processing. This is provided through its own emulation code and internal data set format. The data sets that IAM creates are considered by z/OS as non-VSAM data sets (with a DSORG of either PS or DA).

File Manager recognizes when IAM is present and when displaying information about non-VSAM data sets provides IAM-related information for those data sets that are controlled by IAM.

IAM Entry detail panels contain information describing the non-VSAM data set that IAM uses to emulate a VSAM cluster. This includes the data set name, the catalog in which it belongs, the type of cluster being emulated (KSDS or ESDS), creation and expiry dates of the data set and, if applicable the SMS classes associated with this non-VSAM data set.

The IAM product provides its own form of LISTCAT information which it normally sends to the IAMPRINT data set or, if no IAMPRINT DD is allocated IAM sends the LISTCAT information as TSO messages to the user's screen (when using TSO). File Manager dynamically allocates a temporary data set to the IAMPRINT DD in order to capture and present this information on the same panel as the non-VSAM data set details.

Note: For more information, refer to your IAM documentation.

If you select an IAM entry, the IAM Entry Detail panel is displayed. Figure 82 shows an example of the first part of this panel. You can scroll up and down the list of parameters.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 1 of 32

IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:
IAM100 IAM FILE ANALYSIS - DSN=FMNUSER.TESTMV.IAM.ESDS
       ---------------------------------------------------------------
       FILE FORMAT -- =  ENHANCED  -  FILE STATUS ----------- =  LOADED
       RECORD SIZE -- =     27990  -  FREESPACE - CI% ------- =       0
       CI SIZE ------ =     32768  -  FREESPACE - CA% ------- =       0
       BLOCK SIZE --- =     32760     EXTENDED PE ---------- =    8635 BLOCK
       BLOCK FACTOR - =         1  -  REQUESTED OVERFLOW ---- =         0 RECS

Command ===> _____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up         F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 82. Example: Catalog Services parameters for an IAM entry (first panel)

Defining a new Catalog Entry

You can define a new catalog entry, using an existing entry as a model, or you can define a new entry without a model and specify all of the required fields yourself. If you define a new non-VSAM catalog entry, you must choose to either only define the catalog entry or define the entry and allocate the data set.

To define a new catalog entry using an existing catalog entry as a model:

1. In the Catalog Services panel:
   a. Type the data set name (and optionally the catalog name) for the *existing* entry.
   b. To run the define function in batch, select the **Batch execution** option.
   c. Enter `DEF` on the Command line.
   d. Press Enter.

e. If you are defining an alternate index for a VSAM file (AIX), you can also request that the newly defined AIX is built following the define, or that an existing AIX is deleted, redefined and rebuilt. The define, build and redefine functions are available for online and batch processing.

When you define an alternate index for a VSAM file (AIX), File Manager displays a pop-up panel as shown in Figure 83.
Select the type of AIX processing you want:

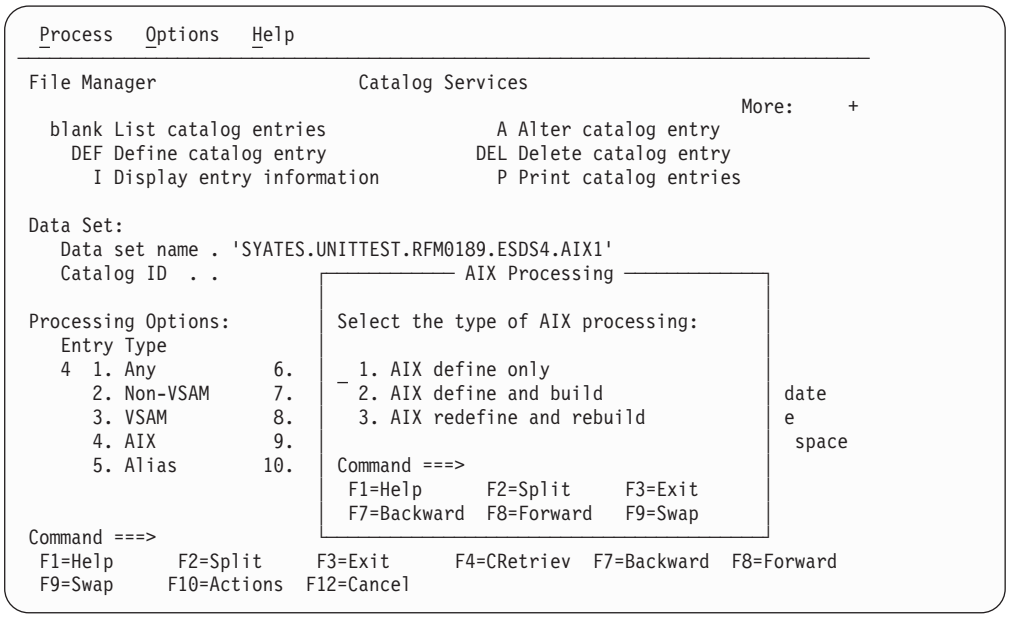


Figure 83. Catalog Services panel

- To define an AIX, select **1. AIX define only**.
- To define and build an AIX, select **2. AIX define and build**.
- To delete, recreate, and then rebuild an existing AIX, select **3. AIX redefine and rebuild**.

A Define panel for the entry type of your existing data set is displayed, containing the information copied from your model. For example, if your existing data set was a VSAM KSDS file, the VSAM Define panel is displayed.

2. Alter the contents of the **Data set name** field to specify a new, unique data set name.
3. For the **Catalog ID** field:
   - If SMS is active on the system, the new value for the catalog ID is copied from the model data set but prefixed with an asterisk (*) to indicate a "comment" value. this value is ignored unless you remove the asterisk.
   - If SMS is not active on the system, the catalog name is copied from the model data set.

   Do one of the following:
   - Blank the field out, to allow the system to assign your data set using the catalog selection order for IDCAMS DEFINE.
   - Remove the asterisk (*) from the existing entry, to assign your data set to the same catalog.

     If the specified catalog is not the same as the related user catalog for this Entry type, a warning message is issued. You can Exit from the message (F3) and change or blank out the catalog name, or you can press Enter to continue using the initial catalog name. If you choose to continue, you need to supply the catalog name in all future searches for the Entry.

- Specify an alternative Catalog ID.

4. For VSAM data sets, alter the component name or names, listed under the Basic Information heading, to unique names suited to your data set.

   OR

   Blank out the component name or names listed under the Basic Information heading. File Manager generates new component names based upon the following AMS rules:

   a. IF the last qualifier of the Entry data set name is CLUSTER, it replaces the last qualifier with DATA for the data component and INDEX for the index component. For example:

      ```
      Cluster name: SALES.REGION2.CLUSTER
      Generated data name = SALES.REGION2.DATA
      Generated index name = SALES.REGION2.INDEX
      ```

   b. ELSE if the cluster name is less than or equal to 38 characters, then append .DATA to the end of the cluster name for the data component and .INDEX for the index component. For example:

      ```
      Cluster name: DEPT64.ASSET.INFO
      Generated data name = DEPT64.ASSET.INFO.DATA
      Generated index name = DEPT64.ASSET.INFO.INDEX
      ```

   c. ELSE if the cluster name is between 39 and 42 characters inclusive, then append .D to the end of the cluster name for the data component and .I for the index component.

      ```
      Cluster name: DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF
      Generated data name = DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF.D
      Generated index name = DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF.I
      ```

   d. ELSE if the name is longer than 42 characters, and the last qualifier is not CLUSTER, use the first (N-1) qualifiers of the cluster, alternate index, or user catalog name up to the first four qualifiers, and append as many 8-character qualifiers as necessary to produce a 5-qualifier name. For example:

      ```
      Cluster name: DIV012.GROUP16.DEPT98.DAILYLOG.DEC1988.BACK
      Generated data name = DIV012.GROUP16.DEPT98.DAILYLOG.TY7RESNO
      Generated index name = DIV012.GROUP16.DEPT98.DAILYLOG.YIIQHNTR
      ```

   After a name is generated, AMS searches the catalog to ensure that the name is unique. If a duplicate name is found, AMS continues generating new names using the format outlined in 4d until a unique one is produced.

5. Alter any of the remaining parameters (copied from the existing entry) to suit your new entry.
6. Press Enter to process your define request.

To define a new catalog entry *without* an existing catalog entry as a model:

1. In the Catalog Services panel:
   a. Type a new, unique data set name (and optionally a Catalog ID).
   b. Select your **Entry Type** option.
   c. To run the define function in batch, select the **Batch execution** option.
   d. Enter `DEF` on the Command line.
   e. If you are defining an alternate index for a VSAM file (AIX), you can also request that the newly defined AIX is built following the define, or that an existing AIX is deleted, redefined and rebuilt. The define, build and redefine functions are available for online and batch processing.

When you define an alternate index for a VSAM file (AIX), File Manager displays a pop-up panel as shown in Figure 83 on page 320. Select the type of AIX processing you want:

- To define an AIX, select **1. AIX define only**.
- To define and build an AIX, select **2. AIX define and build**.
- To delete, recreate, and then rebuild an existing AIX, select **3. AIX redefine and rebuild**.

f. Press Enter.

You might be prompted to further refine your entry type, depending on which Entry Type you selected.

A Define panel for the entry type that you selected is displayed. If this is the first catalog entry you have defined in your current session, the fields in this panel are empty. If you have previously defined a catalog entry with the same type, the field information is pre-filled with the values last used in the panel.

2. Supply the required values and press Enter to process the define request. (File Manager prompts you to complete all of the required fields).
"AIX Entry Detail panel" on page 447
"GDG Entry Detail panel" on page 581
"IAM KSDS Define panel" on page 582
"Non-VSAM Define panel" on page 607
"Non-VSAM Allocate panel" on page 606
"Path Entry Detail panel" on page 612
"VSAM Define panel" on page 719

Altering an existing Catalog Entry

1. Perform either of these actions:
   - In the Catalog Services panel, type a fully qualified data set name (and optionally the catalog name) for the *existing* entry, then enter `A` on the Command line.
   - From the Catalog Services panel, display a list of data set names.

     To rename a VSAM data set (CLUSTER and its corresponding components), enter REN(ame) in the line command field adjacent to the entry you want to change. File Manager displays the VSAM Entry Rename panel for the data type of your data set is displayed, showing the data set information that can be changed. The component names of a VSAM CLUSTER are adjusted automatically based on the newly entered data set name. If a finer degree of control over the component names is required, then the ALTER command may be used to adjust component names manually. To change any additional VSAM parameters, use the ALTER command.
2. In the Data Set List panel, enter `A` in the line command field adjacent to the entry you want to view.

   An Entry Details panel for the data type of your data set is displayed, showing the catalog information that can be altered in editable fields. For example, if your data set is a VSAM KSDS file, the VSAM Entry Details panel is displayed.
3. Alter any of the parameters to suit your requirements.
4. For VSAM data sets, if you alter the data set name and want the component names to match your new name, blank out the component name or names listed under the Basic Information heading. File Manager generates new component names based upon the following ASM rules:
   a. IF the last qualifier of the Entry data set name is CLUSTER, it replaces the last qualifier with DATA for the data component and INDEX for the index component. For example:

```
Cluster name: SALES.REGION2.CLUSTER
Generated data name = SALES.REGION2.DATA
Generated index name = SALES.REGION2.INDEX
```

b. ELSE if the cluster name is less than or equal to 38 characters, then append .DATA to the end of the cluster name for the data component and .INDEX for the index component. For example:

```
Cluster name: DEPT64.ASSET.INFO
Generated data name = DEPT64.ASSET.INFO.DATA
Generated index name = DEPT64.ASSET.INFO.INDEX
```

c. ELSE if the cluster name is between 39 and 42 characters inclusive, then append .D to the end of the cluster name for the data component and .I for the index component.

```
Cluster name: DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF
Generated data name = DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF.D
Generated index name = DEPTABCD.RESOURCE.REGION66.DATA1234.STUFF.I
```

d. ELSE if the name is longer than 42 characters, and the last qualifier is not CLUSTER, use the first (N-1) qualifiers of the cluster, alternate index, or user catalog name up to the first four qualifiers, and append as many 8-character qualifiers as necessary to produce a 5-qualifier name. For example:

```
Cluster name: DIV012.GROUP16.DEPT98.DAILYLOG.DEC1988.BACK
Generated data name = DIV012.GROUP16.DEPT98.DAILYLOG.TY7RESNO
Generated index name = DIV012.GROUP16.DEPT98.DAILYLOG.YIIQHNTR
```

After a name is generated, AMS searches the catalog to ensure that the name is unique. If a duplicate name is found, AMS continues generating new names using the format outlined in 4d until a unique one is produced.

5. Press Enter to process your request.

Related topics

"AIX Entry Detail panel" on page 447
"Catalog Services Data Set List panel" on page 466
"GDG Entry Detail panel" on page 581
"IAM Entry Details panel" on page 583
"Non-VSAM Entry Detail panel" on page 609
"Path Entry Detail panel" on page 612
"VSAM Entry Detail panel" on page 722

Deleting a catalog entry

1. To run the delete function in batch, select the **Batch execution** option on the Catalog Services panel.
2. Perform either of these actions:
   - In the Catalog Services panel, type a fully qualified data set name (and optionally the catalog name) for the *existing* entry, then enter DEL on the Command line.
   - From the Catalog Services panel, display a list of data set names.
3. In the Data Set List panel, enter DEL in the line command field adjacent to the entry you want to view.

   A Delete Entry panel is displayed, showing the data set name, catalog ID and volume serial.
4. Specify your additional delete options, by entering Y or N in the fields:

   Erase Specify Y if the space occupied by the data set is to be overwritten with binary zeros when the data set is deleted; specify N if the space is not

to be overwritten; or leave blank to use the erase option set when the catalog entry was defined or last altered.

Purge Specify Y to delete the catalog entry regardless of the retention period specified, otherwise specify N.

Scratch
Specify Y if the data set is to be removed from the VTOC of the volume on which it resides, otherwise specify N.

This parameter is not applicable to entries which do not have a VTOC entry.

5. Press Enter to process your request.

Note: For non-VSAM data sets, the DEL command does not remove the actual data set, unless you specify Y for Erase, Purge and Scratch.

Recalling catalog entries

When a catalog entry has been migrated or archived, the status is indicated in the **Prim volume** column of the Data Set List panel as follows:

```
 Process   Options   Help
 File Manager                Data Set List                Row 00001 of 00027
Catalog ID ''                                                 Types ALL
        Data Set Name                                  Type  Volume  MV Creat ±
_______ FMN.V11R1M0.S*                                 *     *       *  *
_______ FMN.V11R1M0.SFMNCLIB                           HSM   MIGRAT     09/12
_______ FMN.V11R1M0.SFMNDBRM                           NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNEXEC                           NVSAM D$FM10     09/12
_______ FMN.V11R1M0.SFMNMAC1                           NVSAM D$FM00     09/12
_______ FMN.V11R1M0.SFMNMENU                           NVSAM D$FM19     09/12
_______ FMN.V11R1M0.SFMNMJPN                           NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNMKOR                           NVSAM D$FM03     09/12
_______ FMN.V11R1M0.SFMNMODA                           HSM   MIGRAT     09/12
_______ FMN.V11R1M0.SFMNMODJ                           NVSAM D$FM00     09/12
_______ FMN.V11R1M0.SFMNMODK                           NVSAM D$FM10     09/12
_______ FMN.V11R1M0.SFMNMOD1                           HSM   MIGRAT     09/12
_______ FMN.V11R1M0.SFMNMOD2                           NVSAM D$FM04     09/12
_______ FMN.V11R1M0.SFMNPENU                           NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNPJPN                           NVSAM D$FM16     09/12
_______ FMN.V11R1M0.SFMNPKOR                           NVSAM D$FM13     09/12
Command ===>                                                        Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 84. Migrated catalog entries in Data Set list

To recall these entries:

1. Enter the RECALL line command in the Command area adjacent to the catalog entry.

   Note: A TSO message is displayed, that tells you to press the ATTENTION key to free your TSO session. DO NOT DO THIS.

   ```
    ARC1020I DFSMSHSM IS RECALLING FROM DASD
   DSN=FMN.V3R1M0.DTCU.ASMLLST, YOU MAY
   CONTINUE THE RECALL IN THE BACKGROUND AND FREE
   YOUR TSO SESSION BY PRESSING THE ATTENTION KEY
   ```

2. Wait until TSO indicates that the recall has been completed by displaying

   ***

after the message.

3. Press Enter to return to the File Manager screen.

OR:

1. Enter the command corresponding to the action you wish to take for that data set. For example, to recall a data set to copy it using File Manager's data set copy function, enter `COPY` against the file, or specify the data set in the copy utility as normal, to recall the data set.

Related topics

"Catalog Services Data Set List panel" on page 466

Displaying a Volume Table of Contents (VTOC)

To display or print a list of data sets on one or more disks, you can use the Work with VTOC utility. With this utility, you can list either all the data sets, or a subset of the data sets based on a generic data set name. You can also use Work with VTOC to display or print a volume summary list (VTOC list), or print a combined data set and VTOC list.

See "Display VTOC panel" on page 523 for details.

The information in the data set list includes:

- Data set name
- Volume (VOLSER)
- Disk extents
- File organization information
- Creation, expiration, and referred dates

In online mode, you can enter commands to invoke functions that operate on individual data sets.

See "Display VTOC Data Set List panel" on page 528 for details.

The information in the volume summary list includes:

- Volume information
- VTOC information
- Free space information
- Data set statistics

See "Volume Summary/Selection panel" on page 718 for details.

Note: Work with VTOC shows "never to expire" expiration dates of 99/999 and 99/366 as 9999.999 and 9999.366, respectively. An expiration date that was explicitly specified as 99/365 is also considered "never to expire", and is shown as 9999.365. If you sort the list by expiration date, values beginning with 9999 appear first.

To display the VTOC information:

1. From the Utility Functions menu panel, select option **5. VTOC**. The Display VTOC panel shown in Figure 156 on page 524 is displayed.
2. Specify the disk volume or volumes with which you want to work. You can specify a generic volume serial number, such as D$US1*.

Note: If you specify a pattern that causes a large number of volumes to be accessed, the operation can take a long time.

3. You can limit the data sets that are listed by specifying selection criteria in a number of fields on the Display VTOC panel:

   Data set name
   Specify a generic data set name. If you do not specify a data set name, File Manager uses the default generic data set name of '**' which causes all data sets to be listed or included. If you do not type quotes around the data set name (mask), File Manager prefixes the data set name with your TSO PROFILE PREFIX (usually your user ID). For example, to limit the data sets to just the ones starting with your TSO PROFILE PREFIX, specify the generic data set name of ** (no quotes). Any valid ISPF data set name mask can be used as the generic name.

   Volume serial
   Specify a fully-qualified or generic volume serial number.

   Volume status
   Specify a fully-qualified volume status string of PRIVATE, PRIV/RSDNT, RESIDENT, PUBLIC, or STORAGE.

   Unit Specify a fully-qualified or generic device address.

   Device type
   Specify a fully-qualified or generic device type.

   SMS SG
   Specify a fully-qualified or generic SMS storage group name.

   To select a non-SMS controlled volume in an SMS environment, specify the reserved name, `NONSMS`.

   Processing limit
   Type a "/" to select this option and then specify the maximum number of data sets to be selected for VTOC processing.

4. If required, select the **Batch execution** option. File Manager then displays the Display VTOC Batch Sort Options panel.

   On this panel, select the appropriate sort option. File Manager then generates the batch job stream and the JCL is presented in an Edit session. If necessary, you can change the JCL before submitting the job.

5. To display dates in the format YY/MM/DD (instead of the default format YYYY.DDDD), select the **YY/MM/DD date format (default: YYYY.DDD)** option.

6. By default, for data sets with multiple extents, the display format suppresses the data set name on the second and subsequent extents. To show the data set name for each extent, select the **Repeat data set name for each extent** option.

To update your list view, to include any changes that you have made and also any changes made by other users since you first displayed the list, enter the REFRESH primary command.

To display or print a volume summary list (like the one shown in Figure 85 on page 327):

1. Enter the line command **V** to display the list.
2. Enter the line command **P** to print the list. File Manager then displays the Display VTOC Batch Sort Options panel.

On this panel, select the appropriate sort option. File Manager then produces the requested VTOC listing.

When you specify a generic volume serial number, File Manager displays a list of those volumes that contain data sets with names that match the specified data set name. If you select a data set list command, a data set list with all data sets matching the specified data set name and volume serial is produced.

```
 Process   Options   Help
 File Manager            Volume Summary/Selection           Row 00001 of 00005
Unit    *        DSN       'TYRONED.FMDATA.**'
DevType *        VOLSTATE  ALL         SMS SG *
Volumes 5        Data sets 7           VSAM   3           non-VSAM 4
VOLSER  D$US5*   Trks used N/A         Free   N/A         Utilized N/A
  VOLSER UNIT Dtype SMS-SG   Total  Used   %Used Free trk Tot DSN VSAM   nVSA ±
  D$US5* *    *     *
_ D$US51 E801 3390  PRIMARY  150255 128750    86    21505       1           1
_ D$US53 E81C 3390  PRIMARY  150255 142702    95     7553       1           1
_ D$US55 E929 3390  PRIMARY  150255 141320    94     8935       1           1
_ D$US57 EA52 3390  PRIMARY  150255 127706    85    22549       2      2
_ D$US58 EA55 3390  PRIMARY  150255 131406    87    18849       2      1    1
_
****  End of data  ****

Command ===>                                                    Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 85. Work with VTOC: list of volumes

To see more information about a volume listed on the Volume Summary/Selection panel, enter the line command I (Information) in the **Sel** field next to the volume you are interested in. The I line command displays the Disk Volume Details panel showing detailed information about the volume you selected.

Note: To show data set statistics on the Disk Volume Details panel (under the heading "Data Sets"), the **Limited information for VTOC list** option on the Display VTOC panel (that you used to navigate to this point) must be deselected.

On the Volume Summary/Selection panel you can use the S (Select) line command to select which volume you want to work with. If you want lists of data sets from more than one volume, you can enter the S line command on multiple lines. If you are going to select many volumes, you can use the ALL command to fill in the S on every line for you (you might then, if you wish, deselect any volume by overtyping its S with a blank).

File Manager lists the data sets as shown in Figure 86 on page 328.

```
 Process   Options   Help
 File Manager           Display VTOC Data Set List          Row 00001 of 00055
Unit    *        DSN       'FMN.V11R1M0.S*'
DevType *        VOLSTATE  ALL         SMS SG *
Volumes 10       Data sets 27          VSAM   0           non-VSAM 27
VOLSER  *        Trks used 3752        Free   N/A         Utilized N/A
        Data Set Name                                  Seq Volume Begin CYL-HD  ±
        FMN.V11R1M0.S*                                  *      *
_______ FMN.V11R1M0.SFMNCLIB                             1 D$FM08    32 14
_______ FMN.V11R1M0.SFMNDBRM                             1 D$FM09  8802  4
_______ FMN.V11R1M0.SFMNEXEC                             1 D$FM10  1060  0
_______ FMN.V11R1M0.SFMNMAC1                             1 D$FM00  7885  1
_______ FMN.V11R1M0.SFMNMENU                             1 D$FM19   172 13
_______ FMN.V11R1M0.SFMNMJPN                             1 D$FM09  9025  8
_______ FMN.V11R1M0.SFMNMKOR                             1 D$FM03  6363 10
_______ FMN.V11R1M0.SFMNMODA                             1 D$FM02  2881  0
_______ FMN.V11R1M0.SFMNMODA                             2 D$FM02   109  0
_______ FMN.V11R1M0.SFMNMODA                             3 D$FM02   109 10
_______ FMN.V11R1M0.SFMNMODA                             4 D$FM02   110  5
_______ FMN.V11R1M0.SFMNMODA                             5 D$FM02   111  0
Command ===>                                                    Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 86. Work with VTOC: list of data sets

By using the RIGHT and LEFT commands or corresponding function keys, you can scroll the list to see the other columns.

To print a combined data set and volume summary list report, use the **PC** line command. File Manager produces a summary output for each disk, followed by the list of data sets located on the volume that match the selection mask (if any).

Using line commands with Work with VTOC

For each *line* command (see "Display VTOC Data Set List panel" on page 528), perform either of these actions

- Type the command:
  1. Move the cursor to the beginning of the line that contains the name of the entry that you want to work with.
  2. Type the command (for example, BROWSE). You can type an equal sign (=) to perform the same command for more than one entry.

     You can abbreviate the line commands to their first letter, for example, A instead of ALTER except in the case of EDIT and EXTENTS where you would use the first two letters.
- Select the function from the Process pull-down menu:
  1. Move the cursor to the entry with which you want to work.
  2. Press the Process function key (F6). (If the cursor is outside the list area, you work with the first entry in the list.) The selected line is highlighted in another color. The Process pull-down menu opens below the action bar.
  3. Select an option on the menu to perform a function such as browse or edit.

Related topics

"Manipulating your view of selection lists" on page 28

Browsing AFP Print documents

AFP Print Browse provides a formatted display of an Advanced Function Printing (LIST3820) document. For File Manager to be able to process the AFP file, it must have undergone the final processing performed by ACIF or a similar product. In particular, the file must contain the following records:

- begin document 'd3a8a8'x,
- begin multifunc overlay 'd3a8df'x,
- begin image block 'd3a87b'x
- presentation text 'd3ee9b'x

File Manager formats any readable text using the embedded position information and stores it in a temporary file. Graphical data is not included.

When you select the AFP Print Browse option, the AFP Print Browse panel shown in Figure 115 on page 443 is displayed.

You can specify the range of pages to be formatted and change the viewing size for better formatting results.

You can also use the FIND primary command to find the next occurrence of a character string. For example, `FIND MISPELL` finds the next occurrence of “mispell” in the LIST3820 document.

The formatted data is displayed with additional page separator lines. You can use scroll commands (see “Scrolling to see data” on page 77) and step from page to page through the document.

AFP Print Browse primary commands

You can enter the following primary commands on the command line:

- FIND - see “FIND/FX primary command” on page 769
- LOCATE - see “LOCATE primary command” on page 786
- NEXTPAGE - this command is unique to the AFP Browse panel. For an explanation of its function, see “AFP Print Browse panel” on page 442
- PREVPAGE - this command is unique to the AFP Browse panel. For an explanation of its function, see “AFP Print Browse panel” on page 442

Browsing your user storage

You can use Memory Browse to display your user storage in dump format.

As you review your data, File Manager saves up to 64 browse addresses in a pointer chain. You can move forwards and backwards through this chain, to return to previously visited areas.

To view your user storage:

1. Perform either of these actions:
   - From the Utility Functions menu, select option **8 Storage**.
   - From any panel in File Manager, enter the MB primary command.

     The Memory Browse panel, shown in Figure 198 on page 605, is displayed.
2. Scroll through your storage area, using any of the following techniques.
   - Use the DOWN (F8), UP (F7), TOP and BOTTOM commands. The areas reviewed using these commands are not saved in your pointer chain.

- Enter a location, in hexadecimal format, into the **Browse address** field and then enter the NEXT command. This sets a new browse address and saves the current browse address to the chain.
- Place the cursor onto a pointer value visible on the screen and then enter the NEXT command. This sets a new browse address and saves the current browse address to the chain.
- Enter a FIND command on the Command line. This sets a new browse address and saves the current browse address to the chain.
- Check that your cursor is not on a data field, then enter the NEXT or BACK commands without parameters to move through existing addresses in your chain.

3. You can use the **Search limit** field to restrict the area searched.

Memory Browse primary commands

You can enter the following primary commands on the command line:

- BACK - this command is unique to the Memory Browse panel. For an explanation of its function, see “Memory Browse panel” on page 604
- BOTTOM - see “BOTTOM primary command” on page 742
- CLIPBOARD - this command is unique to the Memory Browse panel. For an explanation of its function, see “Memory Browse panel” on page 604
- DOWN - see “DOWN primary command” on page 760
- FIND - see “FIND/FX primary command” on page 769
- NEXT - this command is unique to the Memory Browse panel. For an explanation of its function, see “Memory Browse panel” on page 604
- TOP - see “TOP primary command” on page 824
- UP - see “UP primary command” on page 826

You can also browse the data saved to the clipboard by the PUT command during an Edit session. To do this, perform either of these actions

- Enter `MB` on the Command line of the Edit panel (the Memory Browse panel is displayed) and enter `CLIPBOARD` on the Memory Browse Command line to display the saved data
- Enter `MB CLIPBOARD` on the Command line of the Edit panel to display the saved data directly.

Viewing load module information

You can use the View Load Module to display or print a list of the symbols (CSECTs, common sections, entry points and ZAPs) in a load module (or program object). This utility also works in batch and with REXX.

To view your Load Module information:

1. From the Utility Functions menu, select option **3.10.1 Load Module Information.**. The View Load Module selection panel, shown in Figure 194 on page 593, is displayed.
2. Specify the load module you want to display, using the fields under the heading **Input**.
3. Use the **Processing Options** to specify the order in which the symbols are displayed, and if you want the output to be displayed on a terminal or spooled to a printer.
4. If desired, select the YY/MM/DD date format option. (The default date format is Julian.)

5. Press Enter. File Manager displays the contents of the specified load module as shown in Figure 193 on page 592.
6. Use the DOWN (F8), UP (F7) TOP and BOTTOM commands to scroll through your data.

Working with WebSphere MQ

From within File Manager, you can access WebSphere MQ queues on the local z/OS system where File Manager is running.

From the Primary Option Menu panel, select option **9 WebSphere MQ**. File Manager displays the Websphere MQ Functions panel:

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Websphere MQ Functions

1  List          List Managers and queues.
2  View          View a WMQ queue.
3  Edit          Edit a WMQ queue.

⋮
```

To access WebSphere MQ queues, from the Websphere MQ Functions panel select option **1 List** to display the Websphere MQ Managers panel.

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
 File Manager               Websphere MQ Managers             Row 0001 of 0004

  SSID   ↓ Name     Active   Code     Reason
  <---+--> <---+--> <---+--> <---+--> <---+----10---+----2--->
_ CSQ1     CSQ1MSTR Active          0 MQRC_NONE
_ CSQ2     CSQ2MSTR Active          0 MQRC_NONE
_ CSQ3     CSQ3MSTR Active          0 MQRC_NONE
_ CSQ4     CSQ4MSTR Active          2 MQRC_Q_MGR_NOT_AVAILABLE
 ****  End of data  ****

⋮
```

The WebSphere MQ Managers panel shows a list of WebSphere MQ queue managers and, for each one, whether File Manager is able to connect to the queue manager.

If File Manager cannot connect to a queue manager, the connection completion code and the MQ mnemonic for the reason code are displayed in the list.

If the required WebSphere MQ libraries are not available to File Manager, the queue managers are shown but the connection status shows "*NOLIB*".

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager               Websphere MQ Managers             Row 0001 of 0004

   SSID   ↓ Name     Active   Code     Reason
   <---+--> <---+--> <---+--> <---+--> <---+----10---+----2--->
   *NOLIB*
_
:
```

- To **display the attributes of a queue manager** from the list of queue managers displayed on the WebSphere MQ Managers panel, enter the I prefix command. File Manager displays the Websphere MQ Manager Information panel.

  For each attribute, you can obtain a description by placing the cursor on the field and pressing the Help function key (typically F1).

  To change any attributes, enter the ALTER primary command (to enable each attribute for editing), type the new values and press the Exit function key (F3) to perform the changes (File Manager issues an ALTER QMGR command).
- To **select a queue manager** from the list of queue managers displayed on the WebSphere MQ Managers panel, enter the S prefix command. File Manager displays the WebSphere MQ Queue List panel showing a list of the available local queues for that queue manager. By default, the local queues are sorted by queue name.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                Websphere MQ Queue List CSQ2     Row 0001 of 0109

   Queue                              ↓ Descr                                  →
   <---+----10---+----2----+----3--> <---+----10---+----2----+----3----+----4->
 _ CICS01.INITQ                       CKTI initiation queue
 _ COOL                               Coolest queue ever
 _ CSQ1.XMIT.QUEUE                    Transmission queue for CSQ1
 _ CSQ2.DEAD.QUEUE                    CSQ2 dead-letter queue
 _ CSQ2.DEFXMIT.QUEUE                 CSQ2 default transmission queue
 _ CSQ3.XMIT.QUEUE                    Transmission queue for CSQ3
 _ CSQ4IVPG.TRIGGER                   WebSphere MQ IVP INITIATION QUEUE
 _ CSQ4IVP1.TRIGGER                   WebSphere MQ IVP INITIATION QUEUE
 _ D3.CSQ2.ANYQ
 _ FMN.TEST.NOGET                     RFM0037 GET NOT ENABLED
 _
:
```

The WebSphere MQ Queue List panel shows the attributes for each queue in the displayed list. You can scroll right or left through the attributes for a queue as indicated by the arrows (→ and ◄).

Note: For a description of these attributes, see the *WebSphere MQ Script (MQSC) Command Reference*.

From the WebSphere MQ Queue List panel, you can invoke various actions by typing one of these prefix commands against a listed queue:

A To add a queue.

File Manager displays the Websphere MQ Queue Information panel with all blank fields allowing you to enter details for the new queue. When

you press the Exit function key (F3), File Manager issues a DEFINE QLOCAL command with the attributes set as entered on the panel. To exit the panel without a define taking place, press PF12 or enter CANCEL.

D To delete the queue.

E To edit the queue.

File Manager displays the Websphere MQ Queue Editor Entry from which you navigate to edit the queue.

I To view information about the queue.

File Manager displays the Websphere MQ Queue Information panel. On this panel, you can change the attributes of the queue by entering the ALTER primary command (File Manager issues an ALTER QLOCAL command).

If you change the name of a queue, File Manager issues a DEFINE QUEUE command (rather than an ALTER command). This is a simple way of duplicating the attributes of a queue under a new name. If you specify the name of an existing queue, File Manager displays an error in a message pop-up panel.

R To discard (reset) all current messages on the queue (File Manager issues an EMPTY QLOCAL command).

V Provided there are messages on the selected queue, File Manager displays the WebSphere MQ Queue Editor Entry panel showing the names of the WebSphere MQ queue and WebSphere MQ manager already entered.

From this point, you can now start a View session (provided GET is enabled for the queue) to display the contents of the selected queue:

1. To include message header information (the message descriptor), select (with the an "/") the **Include descriptors** option.
2. To format the message (or header) contents, similar to a traditional File Manager view, specify a copybook or template in the **Copybook or Template** entry fields.

   To display both the message descriptor information and the message data in a formatted sense, use File Manager's segmented data option on the template.

   File Manager supplies a sample copybook and template in the sample library (SFMNSAM1). The member FMNPMQMD is a PL/I copybook extracted from the Websphere MQ supplied copybook which describes common message header descriptions. This sample may be extended to include your application data layout definition by adding the appropriate PL/I statements, or a %INCLUDE statement.

   Likewise, the member FMNCMQMD is a COBOL copybook which describes common message header descriptions.

   The member FMNTPMQD is the template version of the PL/I copybook and the member FMNTCMQD is the template version of the COBOL copybook.

   Both templates include the identification criteria used to identify each message header type as shown in Table 5 on page 334. These are easily determined by inspecting the copybook.

Table 5. Criteria for identifying message header types

Layout name	PL/I template: add this ID criteria	COBOL template: add this ID criteria
MQCIH	#2=='CIH '	#3=='CIH '
MQDH	#2=='DH '	#3=='DH '
MQDLH	#2=='DLH '	#3=='DLH '
MQIIH	#2=='IIH '	#3=='IIH '
MQMDE	#2=='MDE '	#3=='MDE '
MQMD1	#2=='MD ' & #3=1	#3=='MD ' & #4=1
MQMD2	#2=='MD ' & #3=2	#3=='MD ' & #4=2
MQRFH1	#2=='RFH ' & #3=1	#3=='RFH ' & #4=1
MQRFH2	#2=='RFH ' & #3=2	#3=='RFH ' & #4=2
MQRMH	#2=='RMH '	#3=='RMH '
MQTM	#2=='TM '	#3=='TM '
MQTMC2	#2=='TMC '	#3=='TMC '
MQWIH	#2=='WIH '	#3=='WIH '
MQXQH	#2=='XQH '	#3=='XQH '

3. Press Enter to view the contents of the queue as shown in Figure 87.

```
 Process   Options   Help
------------------------------------------------------------------------------
View             CSQ2:D3.CSQ2.ANYQ               +                Top of 12
       Col 1                                     Record AT TOP     Format CHAR
       ----+----10---+----2----+----3----+----4----+----5----+----6----+----7--
****** ****  Top of data  ****
000001 08062107301100001 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000002 08062107322900001 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000003 08062107342300001 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000004 08062107342300002 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000005 08062107342300003 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000006 08062107342300004 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000007 08062107342300005 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000008 08062107342300006 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000009 08062107342400007 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000010 08062107342400008 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000011 08062107342400009 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
000012 08062107342400010 THIS IS A MESSAGE TEXT          *0000051..*0000061..*0
****** ****  End of data  ****


Command ===>                                                  Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 87. Viewing the contents of a queue

To *view* a WebSphere MQ queues, from the Websphere MQ Functions panel select option **2 View** to display the Websphere MQ Queue Editor Entry panel.

To *edit* a WebSphere MQ queues, from the Websphere MQ Functions panel select option **3 Edit** to display the Websphere MQ Queue Editor Entry panel.

Note: Editing a queue retrieves messages in browse mode into virtual storage for the edit session. In an edit session, you can perform deletes, updates and inserts.

For messages that are updated in the edit session, at save time (including the save and exit performed as a result of pressing the F3 function key) the original message is destructively read from the queue and the updated message PUT to the queue. Messages are read in physical order. This means that updated messages appear in a different physical order than when the queue was first read by the editor. Because the messages are loaded into virtual storage, for queues with a high depth, not all of the messages may be able to be loaded into storage. If this does not provide all of the messages that you are expecting, then it is recommended you use a template to select the particular messages that you are interested in.

Related topics

"WebSphere MQ Functions panel" on page 728
"WebSphere MQ Managers panel" on page 729
"WebSphere MQ Managers Information panel" on page 730
"WebSphere MQ Queue Editor Entry panel" on page 732
"WebSphere MQ Queue Information panel" on page 734
"WebSphere MQ Queue List panel" on page 736

Chapter 8. Tape Specific Functions

When you select Tape Specific Functions, the Tape Specific Functions panel shown in Figure 246 on page 678 is displayed.

Tape Specific Functions allow you to:
- Obtain overall information about tape contents
- Write special tape records
- Erase tapes physically

Specifying tape input or output

When you use File Manager tape options, you can process tape data in one of two ways: basic tape or labeled tape:

Basic tape

With basic tape processing, you work with tape files and tape blocks. The first tape mark after a data record or after the leading tape mark is treated as the end of a file. Basic tape functions cannot process multivolume input tape files. No tape label processing is performed; tape label sets are processed as ordinary tape files.

You can move the tape forward or backward by a number of physical records or a number of tape files. Tape input is read from the current position of the input tape, and tape output is written to the current position of the output tape.

Note:

1. For output tapes, you can position the tape before the last tape mark of a file if you want to extend the tape file.
2. To process a labeled input tape without label processing, position the tape at the first data record; for example, by using Forward Space File (option 4.15.2) to skip the tape label file.
3. If the first record read is a tape mark, File Manager considers it a leading tape mark and continues. To copy a null file correctly, position the tape in front of the tape mark, which is then treated as a leading tape mark. To copy consecutive files (including null files) from a multifile input tape, use Backward Space Record (option 4.15.3) before each copy command.
4. At the end of tape, File Manager writes two tape marks to indicate the end of volume, unloads the full tape, and requests the mounting of another volume on the tape output device. You can thus create an unlabeled multivolume tape file.
5. For tape functions that cross tape marks and for tape output functions, BLP or NL processing is recommended.

Labeled tape

For labeled tape processing, you specify a file ID. An input tape must be positioned in front of the specified label set. (If the file IDs do not match, the function ends with an error.) An output tape must be positioned at the beginning of a tape or immediately after a labeled file.

File Manager processes SL and AL (ANSI) labels. On input, the label format is recognized automatically. For output, use the **TAPELBL** entry field from the Set Processing Options panel to select the label type.

Labeled tape processing is specified when the tape is opened (during interactive allocation). Only the following options are enabled for output label processing:

- Create Tape Data (option 4.7)
- Sequential Data to Tape (option 4.2.7)
- VSAM to Tape (option 4.2.6)

Note: Not all combinations of File Manager keyword values are supported by the underlying operating environment. For information on supported combinations, see your operating environment documentation.

You can specify tape input or output as follows:

1. Specify a ddname for the tape unit.
2. For an output tape, optionally specify a tape mode code, as shown in "Tape density and mode values" on page 340. By default, File Manager does not change density, and uses buffered write mode where applicable.
3. If the ddname is not allocated, you are asked for allocation information.

The first File Manager option that uses a tape opens the tape. If the DDNAME you specify is not allocated, File Manager prompts for the information required by displaying the Tape Allocation panel shown in Figure 88 on page 339. File Manager then dynamically allocates and opens the tape. The tape remains open until you rewind and unload it using Tape Rewind-Unload (option 4.15.6) or exit from File Manager.

- You must specify the unit. Other panel input depends on the specified label value. When you use any tape options to process multiple files (that is, go over tape marks), or any tape output options, BLP or NL processing is recommended. (File Manager basic tape output options do not create labeled tapes).
- If you specify BLP (provided you are authorized), the VOLSER is not verified and any data set name is ignored. If input tape label processing is desired (SL or AL), both the VOLSER and data set name are required and must match. To allocate a scratch tape for output (nonspecific request), omit the VOLSER.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager            Tape Allocation

Tape Allocation:
   DDNAME to use  . . WHICHTAP_
   Volume serial(s) .____________________________________
   Unit . . . . . . . ___          device number, generic, or group name
   Label value  . . . SL_          SL, NL, AL or BLP
   Data set name  . .____________________________________________
   Sequence number  . ____         relative position of data set on tape
   Expiration date  . ________     yyyy.ddd, leave blank if none desired
   Open for write . . NO_          NO or YES
   Disposition  . . . MOD_         OLD, MOD, NEW, or CAT (for NEW,CATLG)









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 88. Tape Allocation panel

When you allocate a tape, you are asked for the following information:

DDNAME to use
The ddname for the tape unit

Volume serial
The actual volser (which is verified if label processing is requested), or an external number to be used for mounting the tape if you do not know the actual volume serial (which requires BLP). For example, you might use Tape Label Display (option 4.8) to find out the actual volume serial number.

To access a volume set, specify up to five volsers in parentheses, for example: `(FMO005 FMO006 FMO007)`. Volume sets are supported by the Tape Label Display (option 4.8), Tape to Labeled Tape (option 4.2.3), and basic tape output functions.

To access a scratch tape (a non-specific volume request), do not enter a volser. The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using `OPEN=OUTIN`, so that the tape can later be used for input if desired.

Unit Tape unit number. You can respond with a device number, device type (generic name), or group name (symbolic name).

Label value
`SL` for standard label processing (the default), `AL` for ANSI label processing, `BLP` for bypass label processing, or `NL` for no labels.

If File Manager is running APF-authorized, BLP is set internally for some functions if the user has the appropriate authority. BLP is reset when the function ends.

Data set name
The name of the tape data set that you want to work with, if label processing is used.

Sequence number
The relative position of a data set on the tape.

Expiration date
The expiration date for the data set.

Open for write
Specify YES to access a tape in read/write mode. The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using OPEN=INOUT, so that the tape can be used for output if desired but labels are not rewritten during open.

Disposition
Data set status. Default is MOD.

OLD The file already exists.

MOD If the file already exists, add records to the end of the file; otherwise create a new file.

NEW Create a new file.

CAT If the step terminates normally, the file is cataloged.

If the File Manager session abends, you might need to use the TSO FREE command to remove any File Manager allocations.

Note: You can also allocate tapes outside File Manager. For example, you might want to use allocation parameters that you cannot specify within File Manager.

If you use options that are not enabled for output label processing, the tape remains open until you rewind and unload it (using Tape Rewind-Unload) or exit File Manager. If you use an option that is enabled for output label processing (Create Tape Data, Sequential Data to Tape, VSAM to Tape), the tape is rewound and unloaded after the file is closed and the task has completed.

Tape density and mode values

Table 6 and Table 7 on page 341 show the density and mode values that are available.

Table 6. Density/mode values for tapes that are not 7-track

Tape Mode (mm)	3480 / 3490 Cartridge Tapes
08	Compaction (IDRC), buffered write mode
28	Compaction (IDRC), immediate write mode
00 or BU	No compaction, buffered write mode
20 or IM	No compaction, immediate write mode
Tape Mode (mm)	**3400 Tapes**
C8 *	800 bpi, dual density
C0 *	1600 bpi, dual density
D0 *	6250 bpi, dual density
Tape Mode (mm)	**3424 / 9348 Tapes**

Table 6. Density/mode values for tapes that are not 7-track (continued)

Tape Mode (mm)	3480 / 3490 Cartridge Tapes
42 *	1600 bpi, buffered write mode
62 *	1600 bpi, immediate write mode
C2 * , 00 or BU	6250 bpi, buffered write mode
E2 * , 20 or IM	6250 bpi, immediate write mode
Tape Mode (mm)	9346 / 9347 Streamer Tapes
50 or LL	IBM 9347, low speed, long gap
60 or LS	IBM 9347, low speed, short gap
90 or HL	IBM 9347, high speed, long gap
30 or HS	IBM 9347, high speed, short gap
00 to FE	IBM 9346, forced streaming mode

Table 7. Tape conversion mode values for 7-track tapes

Tape Mode (mm)	Density (bpi)	Parity	Translation	Conversion
10 *	200	Odd	Off	On
20 *		Even	Off	Off
28 *		Even	On	Off
30 *		Odd	Off	Off
38 *		Odd	On	Off
50 *	556	Odd	Off	On
60 *		Even	Off	Off
68 *		Even	On	Off
70 *		Odd	Off	Off
78 *		Odd	On	Off
90 *	800	Odd	Off	On
A0 *		Even	Off	Off
A8 *		Even	On	Off
B0 *		Odd	Off	Off
B8 *		Odd	On	Off

Note:

1. For mm values marked with an asterisk (*), the tape must be positioned at the load point.
2. If the mm value is omitted, the system default is used.

Tape block size considerations

File Manager can process input and output tape blocks that are larger than 64KB. The theoretical limit for a tape block is 10MB.

If you use tape blocks that are larger than 64KB (32KB for labeled tape processing), be aware that other software may not be able to process such large tape blocks. The hardware that you are using may also limit the block size.

File Manager supports LBI (Large Block Interface) for tape. Note that data sets (which require LBI) can not be allocated while using the ISPF interface. However, existing data set (or data sets) pre-allocated using JCL (in batch) may be processed for both input and output.

Specifying the number of tape files to process

Several File Manager tape options let you specify the number of tape files that you want to work with.

- You can accept the default (to process one tape file).
- You can specify a number of tape files from 1 to 99 999.
- With some tape options, you can specify `EOV` to process all files until the end-of-volume (double tape mark or EOV label).
- With Tape to Tape (option 4.2.1), you can specify `EOT` to copy until the physical end-of-tape. Continuation volumes are not requested.
- With some options, you can specify `EOD` to process all files until a specified delimiter is encountered. First, use the **EOD** entry field on the Set Processing Options (option 0) panel to set an EOD delimiter, then use a tape option with `EOD` as the number of files.

Some tapes (such as program installation tapes) may have multiple tape marks between files. To make sure that all files are processed or to process input tapes beyond the end-of-volume, specify `EOD` or a number greater than the number of files on the tape. To prevent a tape from running off the reel, File Manager stops if it encounters a data check after a tape mark. If reading records does not produce a data check, File Manager processes all records until the physical end of the tape is reached.

Processing tapes with data-check errors

You can process input tapes that have data-check errors. When a data-check error is found, File Manager lets you change, accept, or bypass the tape block containing the error. You can browse the block and be prompted for what to do next with the block.

The blocks that you change are printed, so you have a log of any changes you make.

ASCII translation

Use the **ASCII** entry field on the Set Tape Processing Options panel (option 0.3) panel to translate tape data between ASCII and EBCDIC format for all tape input and output options except the following:

- Initialize Tape (option 4.12)
- Tape Browse (option 4.1)
- Tape Label Display (option 4.8)
- Tape to Labeled Tape (option 4.2.3)
- Tape to Tape Compare (option 4.9)
- Tape Update (option 4.3)

For Tape Label Display, ASCII translation is automatically performed when needed. For Initialize Tape and Tape Browse, you can specify ASCII input or output on the respective panels. For the other functions, translation is not supported by File Manager.

ASCII data is translated to EBCDIC format using the FMNA2E table. EBCDIC data is translated to ASCII format using the FMNE2A table. To translate data from one ASCII format to another ASCII format, customize these tables so that FMNA2E is different from FMNE2A. (For details, see the *File Manager Customization Guide*.)

To translate tape data, set the **ASCII** entry field on the Set Processing Options panel to the appropriate value:

NO Data not translated. This is the default.

IN Translates ASCII tape data to EBCDIC format

OUT Translates EBCDIC data to ASCII format for tape output

BOTH Translates tape data from one ASCII format to another

Note: Translating labels or varying-length format tapes may produce incorrect output.

Exported Stacked Volumes (ESVs)

In this document, movement of logical volumes (and the data they contain) *out of* a Virtual Tape Server (VTS) is referred to as 'export', and movement of logical volumes *into* a VTS is referred to as 'import'. A physical volume managed by a VTS that contains logical volumes that can be removed from the VTS is referred to as an 'Exported Stacked Volume'.

Other terms used in this document in discussions about using File Manager options with Exported Stacked Volumes are:

Internal Stacked Volume. A volume internal to a VTS used to archive logical volumes when the raid array is full. This volume is not in the same format as an Exported Stacked Volume.

logical volume. A tape volume created in a VTS. A logical volume may be physically on raid array in the VTS, on an Internal Stacked Volume, or on an Exported Stacked Volume. The logical volume is equivalent to a physical tape volume in a non-VTS environment.

Requirements for Exported Stacked Volumes

All VTS export volumes are written on devices that support the locate block CCW. File Manager uses the locate block CCW to find the start of a logical volume, therefore the input device for the Exported Stacked Volume must support the locate block CCW.

ESV options

File Manager provides two options for working with Exported Stacked Volumes:

- Exported Stacked Volume Copy (option 4.2.8). See "Exported Stacked Volume Copy (option 4.2.8)" on page 352.
- Exported Stacked Volume List (option 4.14). See "Exported Stacked Volume List (option 4.14)" on page 367.

For further information about using these options, see "EVC (Exported Stacked Volume Copy)" on page 1011 and "EVL (Exported Stacked Volume List)" on page 1013.

The following provides some additional information about Exported Stacked Volume Copy and Exported Stacked Volume List.

- Only Exported Stacked Volumes created with the VTS export function are supported by Exported Stacked Volume Copy and Exported Stacked Volume List. Exported Stacked Volume Copy and Exported Stacked Volume List do not support Internal Stacked Volumes that were not created by the VTS export function. Currently an Internal Stacked Volume is not compatible with an Exported Stacked Volume.

  To use a Exported Stacked Volume with Exported Stacked Volume Copy or Exported Stacked Volume List, the volume must first be removed from the tape library where it was created, and either inserted into another tape library as a normal SL volume, or mounted on a stand alone tape drive.
- For the input Exported Stacked Volume:
  - The input allocation may be SL or BLP. If SL is used, the operating system checks for a correct set of standard label headers (an input Exported Stacked Volume contains SL header records).
  - During the copy process the volume is no longer treated as an SL volume.
  - The volume is rewound at the start of Exported Stacked Volume Copy or Exported Stacked Volume List (for both SL and BLP), the standard label headers internally checked by Exported Stacked Volume Copy or Exported Stacked Volume List, and the volume verified to ensure that it was created by the VTS export function.
- For a valid input Exported Stacked Volume, the Table of Contents (located at the end of an Exported Stacked Volume) is read. The Table of Contents is used to:
  - Verify that requested logical volumes exist on the input Exported Stacked Volume.
  - Index into the input Exported Stacked Volume to locate the start of a logical volume (performance).

Using primary commands with Tape Specific Functions

For some tape functions, you can use *primary commands*.

Primary commands are commands you enter on the Command line or, where applicable, by using a function key. Table 8 shows which primary commands you can use with which tape function.

Table 8. Primary commands you can use with Tape Specific Functions

Command	Tape function		
	Tape Browse[1] (4.1)	Tape Update[1] (4.3)	Tape Record Load[1] (4.4)
BID	✓		
BSF	✓	✓	
BSR	✓	✓	
COPY		✓	✓
FIND	✓	✓	✓
FSF	✓	✓	
FSR	✓	✓	
LOCATE	✓	✓	✓
RD	✓	✓	✓

Table 8. Primary commands you can use with Tape Specific Functions (continued)

Command	Tape function		
	Tape Browse[1] (4.1)	Tape Update[1] (4.3)	Tape Record Load[1] (4.4)
REW	✓	✓	
RFIND	✓	✓	✓
RP	✓	✓	✓
RUN	✓	✓	
TOF	✓	✓	
ZOOM	✓	✓	

Notes:
1. You can also use scrolling primary commands as described in "Scrolling to see data" on page 77

The following sections describe each of the primary commands in Table 8 on page 344.

BID

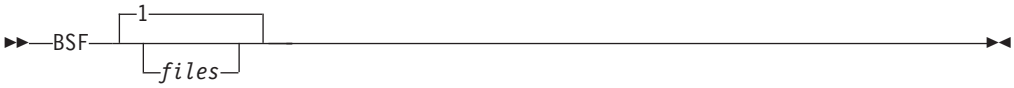
The BID primary command lets you work with the hardware block ID instead of the block numbers within a file. File Manager displays the block ID (including flag byte) for every data block and tape mark. To locate a block with a specific block ID, enter the desired value in the Block-Id field. Enter BID again to return to the block number display. (Working with block IDs may be helpful to move the tape over defective areas which cannot be crossed with other tape positioning commands.)

You can only use BID with Tape Browse (option 4.1).

BSF

The BSF primary command moves the tape backwards a specified number of files.

Syntax



For example, the command `BSF 3` moves the tape backward 3 files.

You can only use BSF with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Backward Space File (option 4.15.1).

BSR

The BSR primary command moves the tape backward a specified number of records.

Syntax

For example, the command `BSR 10` moves the tape backward 10 records.

You can only use BSR with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Backward Space Record (option 4.15.3).

COPY

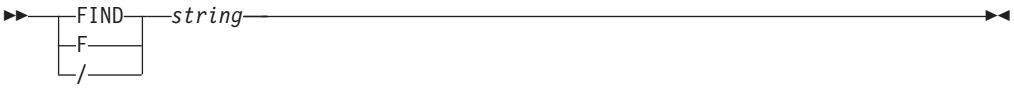
The COPY primary command copies the data denoted by a specified column range within the record replacing the existing data, starting at the first byte of data displayed. COPY is only available when zoomed in on a record (see "Zooming in to see all of a record" on page 71). For further details, see "COPY primary command" on page 755.

You can only use COPY with Tape Update (option 4.3) and Tape Record Load (option 4.4).

FIND

The FIND primary command finds the next occurrence of a character string in the data you are viewing.

Syntax



If the character string is found, it is displayed at the top position. To find the next occurrence of the same string, use the RFIND command, or enter the FIND command with no argument. A message is displayed if the string cannot be found.

The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, "Mixed" matches "MIXED".
- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, "Exact string" matches "exact string".
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, "Exact string" does not match "exact string".
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).

FSF

The FSF primary command moves the tape forward a specified number of files.

Syntax

For example, the command `FSF 2` moves the tape forward 2 files.

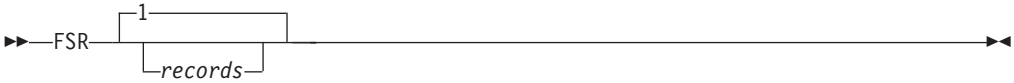
You can only use FSF with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Forward Space File (option 4.15.2).

FSR

The FSR primary command moves the tape forward a specified number of records.

Syntax



For example, the command `FSR 8` moves the tape forward 8 records.

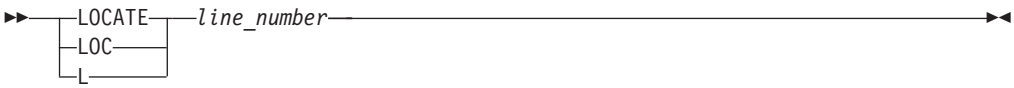
You can only use FSR with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Forward Space Record (option 4.15.4).

LOCATE

The LOCATE primary command positions a specified line number at the top of the display.

Syntax



For example, the command `L 18` positions line 18 at the top of the display.

RD

The RD primary command prints the current record in dump format.

REW

The REW primary command rewinds the tape to the load point.

You can only use REW with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Tape Rewind (option 4.15.5).

RFIND

The RFIND primary command repeats the FIND primary command with the previous search string.

RP

The RP primary command prints the current record in character format.

RUN

The RUN primary command rewinds the tape and unloads it.

You can only use RUN with Tape Browse (option 4.1) and Tape Update (option 4.3).

Note: You can also use Tape Rewind-Unload (option 4.15.6).

TOF

The TOF primary command moves the tape to the beginning of the current file.

You can only use TOF with Tape Browse (option 4.1) and Tape Update (option 4.3).

ZOOM

In CHAR, LHEX, HEX, or TABL display formats, if you want to display all data in a particular record without having to scroll left or right, you can "zoom in" by moving the cursor to the record, then pressing the Zoom function key (F2). To zoom out, press the Zoom function key (F2) again.

You can only use ZOOM with Tape Browse (option 4.1) and Tape Update (option 4.3).

For further details, see "Zooming in to see all of a record" on page 71.

Tape Browse (option 4.1)

Use Tape Browse to browse physical records on a tape. You can display a screen full of records, scroll backward and forward over tape data, search for specific data, and press F2 (Zoom) for a closer look on a single record.

You can specify a range of files to view, and whether to translate ASCII data to EBCDIC for display.

When you select Tape Browse, the Tape Browse panel shown in Figure 89 on page 349 is displayed.

```
Process   Options   Help
-------------------------------------------------------------------------------
File Manager                      Tape Browse

Tapes:  No tapes allocated


Input:
   DDNAME to use ._______        enter new name, or select one from above
   Files . . . . . 1____         number of files or EOV
   Initial BSF . . YES_          NO to start from current tape position
   ASCII data  . . NO_           YES to translate ASCII data for display









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 89. Tape Browse panel

Tape Data Copy Function (option 4.2)

When you select Tape Data Copy Function, the Tape Data Copy Function menu shown in Figure 90 is displayed. This panel allows you to duplicate a tape, copy tape data to another storage device, change the record organization, record format, and block size, recover data from a defective tape, and translate between ASCII and EBCDIC formats.

```
Process   Options   Help
-------------------------------------------------------------------------------
File Manager              Tape Data Copy Functions

1  Tape to tape  Copy tape volume to tape volume
2  Tape reformat Copy and reformat single tape file
3  Multifile     Copy multivolume and multiple tape files
4  To VSAM       Copy from tape to VSAM data set
5  To QSAM       Copy from tape to QSAM data set
6  From VSAM     Copy from VSAM data set to tape
7  From QSAM     Copy from QSAM data set to tape
8  ESV           Copy Exported Stacked Volume









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 90. Tape Data Copy Function panel

Tape to Tape (option 4.2.1)

Tape to Tape copies one or more files from tape to tape without label processing. Tape labels are treated as data. If multiple output volumes are required, the label information may be incorrect.

Also see "Specifying the number of tape files to process" on page 342.

For further information, refer to the online help or "TT (Tape to Tape)" on page 1077.

Tape to Tape Reblocked (option 4.2.2)

Tape to Tape Reblocked copies one file from tape to tape. You can change the record format and reblock the tape file. Tape labels are treated as data.

For further information, refer to the online help or "TTR (Tape to Tape Reblocked)" on page 1080.

Tape to Labeled Tape (option 4.2.3)

Tape to Labeled Tape copies IBM-standard labeled tapes (without requiring you to specify every file ID). Labels read from the input tape are interpreted and copied to the output tape. Copying multivolume and multifile tapes is supported. However, multivolume output tapes are not supported: if the output volume is too short, copying continues on the next tape, but no end-of-volume (EOV) label is written on the full output volume, and no IBM-standard VOL1, HDR1, and HDR2 labels are written on the continuation volume.

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager                Tape to Labeled Tape

Tapes:  No tapes allocated


Input:
   DDNAME to use .________      enter new name, or select one from above
   Type of tape  . SL___        SL or MIXED (mixture of SL and NL files)

Output:
   DDNAME to use .________      enter new name, or select one from above
   Tape mode . . .__            optional recording mode or density code
   Files . . . . . ALL__        number of files to be copied or ALL
   New Volser(s) .______________________________   or * or blank







Command ===> ____________________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 91. Tape to Labeled Tape panel

You can copy all of the files on the tape (by default) or specify the number of files that you want to copy (see "Specifying the number of tape files to process" on page 342). You can start copying at a specific file by positioning the tape at the header label (HDR1) of the desired file. You can also append files on the output volume by positioning the output tape after the last file (EOF2 TM) of the output volume.

You can copy a tape that contains both labeled and unlabeled files by specifying that the type is `MIXED`. Non-labeled files are tolerated and copied as non-labeled files.

Note:

1. Any user labels on the input tape are not copied.
2. File Manager may force bypass label processing (BLP) when switching to the next volume. If BLP is forced, for security reasons the tape is freed (unallocated) when the Tape to Labeled Tape ends.
3. HSM backup tapes should not be copied using Tape to Labeled Tape because File Manager cannot update the HSM internal information regarding the location of the backup data sets.
4. No recataloging or RACF® processing of the data sets takes place because of data set naming conventions on tape.
5. In some error conditions (for example, if unexpired output volumes are used), prompts are sent to the operator console.

For further information, refer to the online help or "TLT (Tape to Labeled Tape)" on page 1060.

Tape to VSAM (option 4.2.4)

Tape to VSAM copies all or selected tape records to a VSAM data set. You can specify the number of records to skip from the beginning of the file and the number of records to copy (see "Specifying the number of tape files to process" on page 342).

You can also use Tape to VSAM to convert a multiple-file tape into a single VSAM ESDS (tape image file), which can be distributed across a network. Specify a number of files other than one, but no tape file ID or start position. See "Copying tape data across a network" on page 354.

Use VSAM to Tape (option 4.2.6) to restore the file to tape.

For further information, refer to the online help or "TV (Tape to VSAM)" on page 1081.

Tape to Sequential Data (option 4.2.5)

Tape to Sequential Data copies all or selected tape records to a sequential data set on disk, tape, or SYSOUT. You can specify the number of records to skip from the beginning of the file and the number of records to copy (see "Specifying the number of tape files to process" on page 342).

You can also use Tape to Sequential Data to convert a multiple-file tape into a single sequential data set (tape image file) which can be distributed across a network. Specify a number of files other than one, but no tape file ID or start position. See "Copying tape data across a network" on page 354.

Use Sequential Data to Tape (option 4.2.7) to restore the file to tape.

For further information, refer to the online help or "TS (Tape to Sequential Data)" on page 1074.

VSAM to Tape (option 4.2.6)

VSAM to Tape copies all or selected records from a VSAM data set to tape. Bypass label processing (BLP) is recommended for this option.

You can specify the number of records to skip from the beginning of the input data set, or the key or slot value to start processing. You can also specify the output record format and block size.

If you used Tape to VSAM to copy a multifile tape to a single VSAM ESDS (tape image file), you can use VSAM to Tape to copy the file back to tape. The resulting tape is an exact copy of the original tape. See "Copying tape data across a network" on page 354.

For further information, refer to the online help or "VT (VSAM to Tape)" on page 1089.

Sequential Data to Tape (option 4.2.7)

Sequential Data to Tape copies all or selected records from a sequential data set to tape. Bypass label processing (BLP) is recommended for this option.

You can specify the number of records to skip from the beginning of the input data set. You can also specify the output record format and block size.

If you used Tape to Sequential Data to copy a multifile tape to a sequential data set, you can use Sequential Data to Tape to copy the file back to tape. The resulting tape is an exact copy of the original tape. See "Copying tape data across a network" on page 354.

For further information, refer to the online help or "ST (Sequential Data to Tape)" on page 1056.

Exported Stacked Volume Copy (option 4.2.8)

When you use Exported Stacked Volume Copy (option 4.2.8), File Manager displays the Exported Stacked Volume Copy panel as shown in Figure 92 on page 353.

```
Process   Options   Help
_____________________________________________________________________________

File Manager             Exported Stacked Volume Copy

Tapes:  No tapes allocated


Input:
   DDNAME to use .________      enter new name, or select one from above
   Logical vol(s)_____________________________________
                                enter logical volume(s) to be copied

Output:
   DDNAME to use .________      enter new name, or select one from above
   Tape mode . . .__            optional recording mode or density code









Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 92. Exported Stacked Volume Copy panel

The following points relate to Exported Stacked Volume Copy:

- Copying from an Exported Stacked Volume to disk is not currently supported.
- Copying part of a logical volume or certain files from a logical volume contained on an Exported Stacked Volume is not currently supported. The complete logical volume must be copied.
- For the output volumes:
  - The allocation may be SL, NL, or BLP. If SL is used. the operating system checks for a correct set of standard label headers.
  - During the copy process the volume is no longer treated as an SL volume.
  - The volume is rewound at the start of Exported Stacked Volume Copy (for SL, NL, and BLP).
  - If the volume contains a VOL1 header, the VOLSER in the VOL1 is saved and used to change the VOL1, HDR1, and EOF1, of records from the logical input volume prior to writing to the output volume. This results in a change of VOLSER for the logical volume being copied.

    To keep the VOLSER on the physical output volume the same as the VOLSER on the logical input volume, initialize the physical output volume with the same VOLSER as the logical volume prior to starting Exported Stacked Volume Copy.
  - If the output volume contains no VOL1 record initially, then the VOLSER of the logical input volume is retained, but no update of tape management systems occurs.
- Exported Stacked Volume Copy extracts a block of data from an Exported Stacked Volume and writes the physical block of data to the output volume. A block of data may be user data or HDR, or EOF records. There is no open or close performed for each file on the logical volume. Hence there is no interface to tape management systems, and the catalog information for individual data sets on the volume is not updated.
- After the Table of Contents has been read, the Exported Stacked Volume is then rewound before the copy process starts.

- Requested logical volumes are copied to the physical volumes. The copies are in the same order as the requested list of volumes.
- Up to five serial numbers of logical volumes contained on an input Exported Stacked Volume can be processed within a single File Manager command.

  Each logical volume requested is copied to a separate single physical output volume. Therefore, a request for four logical volumes to be copied requires four physical output volumes. There must be a one-to-one correspondence of logical input volumes to physical output volumes.

  Currently this utility does not output a single logical input volume to more than a single volume. Therefore, if the output volume is too short to hold the logical input volume, the copy stops.

For further information, refer to the online help or "EVC (Exported Stacked Volume Copy)" on page 1011.

Copying tape data across a network

To send the contents of a tape across a network:

1. Use Tape to VSAM (option 4.2.4) or Tape to Sequential Data (option 4.2.5) to copy all the files on the tape, including tape marks, to a VSAM entry-sequenced data set or a sequential data set. See "Specifying the number of tape files to process" on page 342.

   Each tape mark is converted to an 80-character record that has the format:

```
FILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFILEMTMFIL
```

2. Transfer the file or data set across the network, using a file transfer program, such as the IBM File Transfer Program.
3. At the other end, use the VSAM to Tape (option 4.2.6) or Sequential Data to Tape (option 4.2.7) to copy the data set to a tape. File Manager converts the tape-mark records back to tape marks on the output tape.

Tape Update (option 4.3)

Use Tape Update to make interactive changes to tape blocks in a tape file, while copying it to another tape. (The data on the input tape remains unchanged.) At any time, both tapes are positioned at the same block within the file. When you end Tape Update, the remaining data (up to the limit of files specified) is copied.

To update a tape block, perform the following steps:

1. Select the Tape Update (option 4.3). The Tape Update panel shown in Figure 93 on page 355 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                      Tape Update

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Files . . . . . 1____        number of files or EOV









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 93. Tape Update panel

2. Select the first block that you want to update. Either move the cursor to the block, or use any of the positioning options that are available in Tape Positioning Functions (option 4.15).
3. Press the Zoom function key (F2). This gives you a detailed view of the selected record.
4. Update the block in any of these ways:
   - Overtype the length field with a new record length.
   - Overtype the existing text with new text.
   - Enter a column number in the **Col** field, and use the GET or COPY command to overwrite data in the block, starting at the specified column.

     For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns of the current block that you want to copy.

     For example, if the value in the **Col** field is 5 and you enter `COPY 10-12`, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter `COPY 10`, the character in column 10 replaces the character that was previously in column 5.
5. Use the Exit function key (F3) to save your changes or press the Cancel function key (F12) to discard the changes. A log of your changes is printed. You can select another block to update or leave the function.

Special considerations

There are several differences between Tape Update and the other update functions:

- Tape Update *copies* tape blocks from one device to another device, with the changes that you specify. If you make a mistake, you still have the original data on the input tape as a backup.
- You can copy and update several consecutive tape files by specifying the number of files that you want to update. To update all the files until a double tape mark or an EOV label is reached, specify `EOV` in the **Files** entry field.

- When you use Tape Update, both the input tape and the output tape are backspaced to the nearest tape mark (if they are not already positioned immediately after a tape mark).
- Blocks are copied dynamically, as you move through the files. For example:
  1. You select Tape Update and specify an input tape and an output tape. You accept the default (to update only one tape file).

     One screen of data is displayed while copying to the output tape.
  2. You move forward 50 blocks.

     The first 50 blocks are written to the output tape.
  3. You move backward 10 blocks.

     Both tapes move back 10 blocks.
  4. You change the current block (block 41).

     The changed block is written to the output tape.
  5. You end Tape Update.

     All blocks from block 42 to the end of the file are written to the output tape.
- Because blocks are copied dynamically, *always* make your updates in ascending order, starting at the beginning of the file. Simply scrolling backwards does not cause any updates to be lost. However, if you *update* a block (with a lower block number than previously changed blocks), all previous changes to blocks with a higher number are lost. Also, if you rewind the tapes, *all* updates are lost.

  For example, if you change block 50, then change block 40, then end Tape Update, this is what happens:
  1. Blocks 1 to 50 are copied.
  2. The changed block 50 is written.
  3. The changed block 40 is written.
  4. Blocks 41 to the end of the file are copied. The change that you made to block 50 is lost.
- Throughout the update process, the two tapes should remain in synchronization. If an input/output error causes the tapes to get out of synchronization, use Tape Rewind (option 4.15.5) to rewind the tapes and restore synchronization. (Note, however, that rewinding the tapes causes all previous updates to be lost.)

Tape Update does not support ASCII conversion.

Tape Record Load (option 4.4)

When you select Tape Record Load, the Tape Record Load panel shown in Figure 94 on page 357 is displayed. This panel allows you to copy a specified number of tape records to another tape and alter selected blocks.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                    Tape Record Load

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 94. Tape Record Load panel

Like Tape Update (option 4.3), Tape Record Load copies from one tape file to another tape file with the change that you specify. If you make a mistake, you still have the original data on the input tape as a backup.

Unlike Tape Update, Tape Record Load lets you start and stop copying at any location within a file. You can therefore make a copy of a file with certain records removed. By using Tape Record Load repeatedly, you can combine data from two or more input files, or rearrange the data from a single input file.

For further information, refer to the online help.

Tape Print (option 4.5)

Use Tape Print (option 4.5) to print tape records in character or hexadecimal format. You can specify the record format, the record length, the number of files that you want to print, and the maximum number of records to print per file.

When you select Tape Print, the Tape Print panel shown in Figure 95 on page 358 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                        Tape Print

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Print Options:
   Print format  . CHAR_        CHAR or HEX
   Record format . U__          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . _______      if deblocking desired, fixed format only
   Records . . . . ALL_____     limit number of records to be printed
   Files . . . . . 1____        number of tape files, EOV, or EOD









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 95. Tape Print panel

You can control the format of the printed output using the **DUMP** entry field of Set Print Processing Options panel (see "Set Print Processing Options panel (option 0.1)" on page 658).

Printing starts with the block following the current position and with the first block of each subsequent file. A leading tape mark is printed, if it exists. If you do not specify the number of blocks or files to be printed, one file is printed (up to the next tape mark).

For further information, refer to the online help or "TP (Tape Print)" on page 1063.

Controlling print output

Several options on the Set Processing Options (option 0) panel affect the printed output:

DATAHDR
: Whether the output contains a data header for each record

DUMP
: Which format (updown or across) is used for hexadecimal print output

HEADERPG
: Whether the output contains a header page at the beginning

PAGESIZE
: How many lines per page the output has

PAGESKIP
: Whether output from each function starts on a new page

PRINTDSN
: The print data set where print output is directed when the PRINTOUT print option is set to SYSOUT=*c*

PRINTLEN
: How many columns wide the output is

PRINTOUT
: Where the output is sent

PRTTRANS
: Whether non-printable characters are translated to blanks

RECLIMIT
Which part of each record to print

For more information, see "Set Print Processing Options panel (option 0.1)" on page 658.

Tape Map (option 4.6)

Use Tape Map (option 4.6) to print a summary (map) of the data contained on a tape. You can produce a map of the entire tape, or of a specific number of tape files.

When you select Tape Map, the Tape Map panel shown in Figure 96 is displayed.

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager                        Tape Map

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Print Options:
   Files . . . . . 1____        number of tape files, EOV, or EOD
   Blocks  . . . . 1_______     number of blocks
   Print format  . HEX_         CHAR or HEX









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 96. Tape Map panel

Tape Map prints from each file the number of blocks specified (in hexadecimal format or character format), followed by a summary of the space that the files take up on the tape. The output is affected by several options within Set Processing Options (option 0). These are described in "Controlling print output" on page 358.

Figure 97 on page 360 shows sample output from Tape Map.

```
:
IBM File Manager for z/OS                                                    6/09/2000 (2000.161) 12:56  Page 1
$$FILEM TMP INPUT=TAPEIN
* * * * Device 0580, TAPEIN, VOLSER=0GB220, 9 Track, 1600-BPI,  Ring=in   Tape positioned at load point       * * * *
BLK    1   DATA   80    CHAR VOL10GB220                                GA2972A
                        ZONE EDDFFCCFFF44444444444444444444444444444444CCFFFFC44444444444444444444444444444444
                        NUMR 56310722200000000000000000000000000000000712972100000000000000000000000000000000
                           01...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
File 00001 has 240 bytes in 3 blks, min blk=80, max blk=80
 This file is approximately 0.5 feet (0.1 meters)
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
Total data mapped: 240 bytes, 3 blocks, 0.5 feet (0.1 meters)
3 block(s) mapped
:
:
```

Figure 97. A tape map: output from Tape Map (option 4.6)

For further information, refer to the online help or “TMP (Tape Map)” on page 1061.

Create Tape Data (option 4.7)

When you select Create Tape Data, the Create Tape Data panel shown in Figure 98 is displayed.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                  Create Tape Data

 Tapes:  No tapes allocated


 Input:
  DDNAME to use . ________              enter new name, or select one from above
  Tape mode . . . __                    optional recording mode or density code

 Output:
  Records . . . . ________              number of records
  Record length . 50
  Fillchar  . . . _____                 char or hex value, AN, BIN, or RAND
  Sequence field position  . _______    if sequence field desired
  Sequence field length  . . 8          length from 1 to 9
  Sequence field increment . 10_______  increment value
  Record format . ___                   F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
  Block size  . . _______               required for blocked output


 Command ===> ___________________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 98. Create Tape Data panel

Use Create Tape Data to write simple test records to a tape file. You specify the tape that is to contain the records, the number of records, and the record length. You can also specify a fill character or fill pattern, a sequence field, a record format, and a block size.

For further information, refer to the online help or “BT (Create Tape File)” on page 838.

Tape Label Display (option 4.8)

If a tape is labeled (or if you want to find out if it is labeled) use Tape Label Display (option 4.8) to print the tape labels. This lets you determine the volume serial number of the tape, or determine which data sets are on the tape.

When you select Tape Label Display (option 4.8), the Tape Label Display panel shown in Figure 99 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Tape Label Display

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above
   Label printout  LONG_        specify SHORT to only print VOL1 and
                                first HDR label set









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 99. Tape Label Display panel

You can use Tape Label Display even if you do not have access to the data on the tape, because only the tape labels are read. To mount and process tapes with unknown volsers and labels, the Tape Label Display turns on BLP internally. If your installation does not allow BLP usage but File Manager is running APF-authorized, File Manager forces BLP for this function; for security reasons, the tape is freed (unallocated) when Tape Label Display ends.

Figure 100 on page 362 shows an example of the default output from Tape Label Display. You can also request a shorter report that lists only the VOL1 label and the header labels for the first data set.

```
⋮
IBM File Manager for z/OS                                                    6/09/2000 (2000.161) 12:59  Page 1
$$FILEM TLB INPUT=TAPEIN
* * * * Device 0580, TAPEIN, VOLSER=0GB220, 9 Track, 1600-BPI,  Ring=in                                  * * * *
      VOL1 label =           VOL10GB220                              GA2972A
-------------------------------------------------------------------------------------------------------------------
Data set 0001                1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
      HDR1 label =           HDR1FMNUSER.TEST.FILE 0GB22000010001     000147000000000000IBM OS/VS 370
      HDR2 label =           HDR2V327600008440FMNUSER /ISPFPROC    B  800000
               * Tape mark
               *            1 data block(s) skipped
               * Tape mark
      EOF1 label =           EOF1FMNUSER.TEST.FILE 0GB22000010001     000147000000000001IBM OS/VS 370
      EOF2 label =           EOF2V327600008440FMNUSER /ISPFPROC    B  800000
               * Tape mark
-------------------------------------------------------------------------------------------------------------------
               * Tape mark
-------------------------------------------------------------------------------------------------------------------
End of volume (double tape mark) reached, label summary follows
Label summary for volume 0GB220:
      Data Set Name        Blocks BLKSIZE LRECL RECFM Created Expires Security
   1  FMNUSER.TEST.FILE         1   32760    84   VB  2000147             NO
⋮
```

Figure 100. A tape label summary: output from Tape Label Display

For further information, refer to the online help or “TLB (Tape Label Display)” on page 1059.

Tape to Tape Compare (option 4.9)

Use Tape to Tape Compare to compare, byte by byte, the data on one tape with the data on another. Tape to Tape Compare is particularly useful if you have made a copy of a tape, and want to check that the duplicate is exactly the same as the original.

When you select Tape to Tape Compare, the Tape to Tape Compare panel shown in Figure 101 is displayed.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                Tape to Tape Compare

 Tapes:  No tapes allocated


 Input:
    DDNAME to use . ________     enter new name, or select one from above

    DDNAME to use . ________     enter new name, or select one from above

    Files . . . . . 1____        number of tape files, EOV, or EOD









 Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 101. Tape to Tape Compare panel

You can compare an entire tape, or a specific number of files. Tape to Tape Compare stops processing as soon as it finds the first difference. Be sure to position the tapes as needed before starting the comparison.

To compare all the files on the tapes, enter EOV in the **Files** entry field. If you do not specify a value for **Files**, only one file (from the current block to the next tape mark) is compared.

When a difference is found, File Manager lists:
- The two records that do not match
- The lengths of these records on the two tapes
- The location of the first different byte
- The number of blocks and bytes compared

If no difference is found, File Manager lists the number of blocks and bytes compared.

Note: An input/output error is considered a difference, even if the same error occurs on both tapes.

For further information, refer to the online help or "TTC (Tape to Tape Compare)" on page 1079.

Tape Record Scan (option 4.10)

Use Tape Record Scan to find specific data on a tape.

When you select Tape Record Scan, the Tape Record Scan panel shown in Figure 102 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Tape Record Scan

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above
   Record format . U__          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . _______      if deblocking desired, fixed format only

Scan Options:
   Pattern scan  . NO_          YES or NO
   Scan position . 1____        byte position within record to start at
   Scan argument . ____________________________________________________
   Number of hits  1_______     or ALL to list all scan hits in the file
   Records . . . . ALL_____     limit number of records to be scanned




Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 102. Tape Record Scan panel

When using Tape Record Scan:
- You identify the tape device (**DDNAME to use**)

- You can specify a record format (**Record format**) and, if the records are fixed-length, the record length (**Record length**), to deblock the tape records while searching.
- You can specify where in each record you want to search (**Scan position**). The default is to search for strings that start in column 1.
  - To search for strings that start in column *n*, enter `NO` in the **Pattern scan** entry field and enter *n* in the **Scan position** entry field.
  - To search for strings anywhere in the record, enter `YES` in the **Pattern scan** entry field.
  - To search for strings anywhere from column *n* to the end of the record, enter `YES` in the **Pattern scan** entry field and enter *n* in the **Scan position** entry field.
- The string to search for has the same format as used by the FIND command in all functions, as well as for Disk Record Scan.
  - A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, `'Mixed'` matches `'MIXED'`.
  - A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`
  - `C` followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `'Exact string'` does not match `'exact string'`
  - `X` followed by a hexadecimal string enclosed in quotation marks. For example, `X'C1C2'`.
- You can specify the number of occurrences of the string that should be found before the search stops. The default is to stop after the first occurrence.
- You can specify the number of records to search through. The default is to scan to the end of the file.

Tape Record Scan stops after finding the specified number of occurrences of the string. The tape is positioned immediately after the record containing the last occurrence. If the string is not found, the tape stops after the next tape mark.

For further information, refer to the online help or “TRS (Tape Record Scan)” on page 1072.

Write Tape Mark (option 4.11)

When you select Write Tape Mark, the Write Tape Mark panel shown in Figure 103 on page 365 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                    Write Tape Mark

Tapes:  No tapes allocated


Output:
   DDNAME to use .________       enter new name, or select one from above
   Tape mode . . .__             optional recording mode or density code
   Tape marks  . . 1___          number of tape marks to be written









Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 103. Write Tape Mark panel

Write Tape Mark allows you to write one or more tape marks at the current position.

For further information, refer to the online help or "WTM (Write Tape Mark)" on page 1092.

Initialize Tape (option 4.12)

Use Initialize Tape (option 4.12) to initialize a labeled or unlabeled tape. The tape is rewound, and the appropriate header information is written.

When you select Initialize Tape, the Initialize Tape panel shown in Figure 104 on page 366 is displayed.

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Initialize Tape

Tapes:  No tapes allocated


Output:
   DDNAME to use .________      enter new name, or select one from above
   Tape mode . . .__            optional recording mode or density code
   Volume serial ._____         volume identifier or blank for NL tape
   Data set name ._____________________________________________
   ASCII format  . NO_          YES or NO
   Scale percent . 0_           optional scale percentage or 0 for all









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 104. Initialize Tape panel

To initialize with labels, File Manager writes a volume label set in either IBM standard format or ANSI type 3 or 4 (ASCII) format. For NL initialization, File Manager writes a single tape mark.

Specify a DDNAME for the tape. If the DDNAME is not allocated, you are asked for allocation information. To check existing labels on the output tape before creating the new label set, specify the appropriate label value on the tape allocation panel. Specifying BLP initializes the tape unconditionally.

When Initialize Tape has finished, the tape remains positioned after the label set.

It is recommended that you unload the tape and reallocate it with the new volume serial number, to make your system aware of the new volume serial number.

You can initialize a tape with ANSI type 3 or 4 labels to be used with ASCII files. You can also specify a file identifier to be used for the file header label. Omit the volser to initialize an NL tape.

You can add a scaling capacity to the tape provided the function is supported by the hardware. The scaling capacity limits the amount of the tape actually used.

For further information, refer to the online help or “INT (Initialize Tape)” on page 1027.

Erase Tape (option 4.13)

When you select Erase Tape, the Erase Tape panel shown in Figure 105 on page 367 is displayed. This panel allows you to erase a tape from the current position until end-of-tape (EOT).

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                        Erase Tape

Tapes:  No tapes allocated


Output:
   DDNAME to use ._________     enter new name, or select one from above
   Tape mode . . .__            optional recording mode or density code









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 105. Erase Tape panel

Erase Tape:
1. Writes two tape marks at the current tape position
2. Erases the remaining tape data to the end of the tape (data security erase)
3. Rewinds the tape

Erase Tape can be used only with tape units that support the data security erase I/O command code, such as the IBM 3400 series.

For further information, refer to the online help or “ERT (Erase Tape)” on page 1010.

Exported Stacked Volume List (option 4.14)

When you select Exported Stacked Volume List, the Exported Stacked Volume List panel shown in Figure 106 on page 368 is displayed.

```
Process   Options   Help
-------------------------------------------------------------------------------
File Manager             Exported Stacked Volume List

Tapes:  No tapes allocated


Input:
   DDNAME to use .________       enter new name, or select one from above
   Type of listing SHORT         enter LONG to list logical volume details
   Logical vols(s)_____________________________________
                                 For LONG, enter logical volume(s)
                                 to be listed
                                 or
   Logical start . 0___          enter logical start volume
   Logical end . . 0___          enter logical end volume









Command ===> ____________________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 106. Exported Stacked Volume List panel

Exported Stacked Volume List prints the table of contents (TOC) from an Exported Stacked Volume.

File Manager first verifies that the volume loaded was created by the VTS export function. it then forward-spaces the volume to the table of contents and prints a listing of the table of contents. The sequence printed is the same as the sequence of the logical volumes on the Exported Stacked Volume.

You can obtain a more detailed list of the table of contents by entering `LONG` in the **Type of listing** entry field. The LONG listing type shows standard label header and trailer information, and the location of tape marks.

For further information, refer to the online help or "EVL (Exported Stacked Volume List)" on page 1013.

Tape Positioning Functions (option 4.15)

When you select Tape Positioning Functions, the Tape Positioning Functions menu shown in Figure 107 on page 369 is displayed.

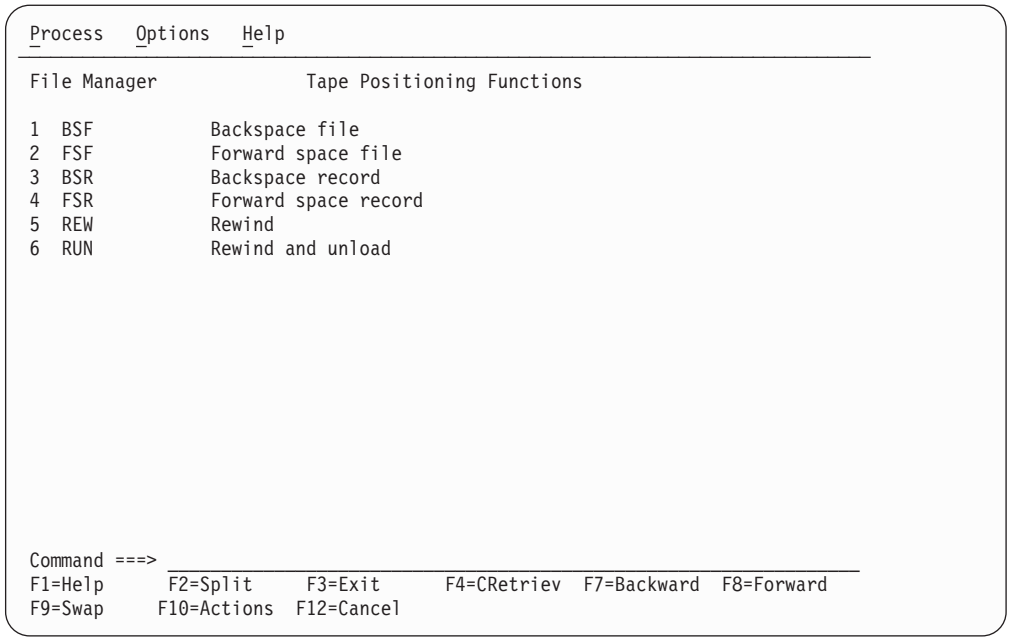


Figure 107. Tape Positioning Functions menu

The Tape Positioning Functions menu allows you to select one of the options shown below. For further information, refer to the online help or the File Manager option indicated.

- Backward Space File (see "BSF (Backward Space File)" on page 837)
- Forward Space File (see "FSF (Forward Space File)" on page 1025)
- Backward Space Record (see "BSR (Backward Space Record)" on page 837)
- Forward Space Record (see "FSR (Forward Space Record)" on page 1026)
- Tape Rewind (see "REW (Tape Rewind)" on page 1044)
- Tape Rewind-Unload (see "RUN (Tape Rewind-Unload)" on page 1044)

Note: To correctly position a tape, you should know the format of the tape. Refer to *z/OS DFSMS: Using Magnetic Tapes*.

Chapter 9. Disk/VSAM Data Functions

Use the options on this menu to work directly with data, rather than working with records or templates. Because you are directly working on the data, use these options with care.

Processing disk records

You can process physical disk extents *within a data set*.

Each disk function works with a single volume. If a data set is split across two or more volumes, disk functions apply to the volser that you specify or (if the data set is cataloged) the first volume listed for the data set in the catalog.

Specifying disk input and output

All disk functions normally work with a disk data set.

To work with a disk data set, specify the data set name. If the data set is not cataloged, specify the volser of the disk.

Specifying disk extents

In most disk functions, you can specify a disk address or a disk address range. You can specify these addresses in the form *cylhd*, where the last 2 digits represent the head value and the other digits represent the cylinder.

If the **CYLHD** entry field of the Set Processing Options panel has the value ABSOLUTE, each *cylhd* value represents an absolute address (counting from the beginning of the physical disk). If the entry field has the value RELATIVE, each *cylhd* value represents an address relative to the beginning of the data set. (See "Set System Processing Options panel (option 0.2)" on page 663 for more information.)

The entry panels for the Disk/VSAM Data Functions tell you whether absolute or relative addresses are in effect.

Disk Browse (option 5.1)

Use this option to display the data structure on a physical disk.

You can change the top record of the display by overtyping the cylinder, head and record numbers displayed in the top left of the panel. You can also overtype the column number, to reposition the data. File Manager changes the column number when it repositions a record after a FIND primary command.

Disk Browse primary commands

The primary commands that you can use with the Disk Browse panel are:

- Standard scrolling forwards and backwards
- FIND
- RD
- RFIND
- RP
- ZOOM

These commands follow the standard browse primary command syntax explained in Chapter 15, “Primary commands,” on page 739, except for FIND. Because of the display constraints when browsing disks, RD always prints the current record in dump format, and RP always prints the current record in character format.

FIND

You use this primary command to search for specific data.

Syntax

Parameters

string Search string. Can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, `Mixed` matches `MIXED`.
- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).

If the string is found, it is displayed at the top position. The **Col** field indicates the position in the record. To find the next occurrence of the same string, press the RFind function key (F5) or enter the FIND command with no argument. A message is displayed if the string cannot be found. (See Chapter 15, “Primary commands,” on page 739 for more information about browse commands.)

Disk Track Edit (option 5.2)

Use this option to create, modify and manipulate records on a disk track.

You must specify the device, the data set name, or both and optionally the physical location of the track.

If you specify a track, File Manager presents the data immediately for editing.

If you do not specify a track, File Manager displays the selected extent in browse mode.

To edit and update a record:

1. Select the record that you want to update. Either move the cursor to the record, or use any of the locate commands that are available in browse functions.

2. Press the Zoom function key (F2). This gives you a detailed view of the selected record.
3. Update the record by:
   - Overtyping the existing text with new text. (You can overtype the characters or the hex.)
   - Entering a column number in the **Col** field, and use the GET or COPY command to overwrite data in the record, starting at the specified column.

     For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns of the current record that you want to copy.

     For example, if the value in the **Col** field is 5 and you enter `COPY 10-12`, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter `COPY 10`, the character in column 10 replaces the character that was previously in column 5.
   - Entering a standard primary edit command.
4. Use the Exit function key (F3) to save your changes or the Cancel function key (F12) to discard the changes. A log of your changes is printed. You can now select another record to update.

Disk Track Edit primary commands

The primary commands that you can use with the Disk Track Edit panel are:

- Standard scrolling forwards and backwards
- ADDKEY
- BOUNDS
- CHANGE
- COPY
- DELETE
- GET
- EXCLUDE
- FIND
- RCHANGE
- RD
- RECOVER
- RFIND
- RP
- ZOOM

These commands follow the standard edit primary command syntax explained in Chapter 15, “Primary commands,” on page 739, except for ADDKEY, CHANGE, EXCLUDE, and FIND. Because of the display constraints when browsing disks, RD always prints the current record in dump format, and RP always prints the current record in character format.

ADDKEY

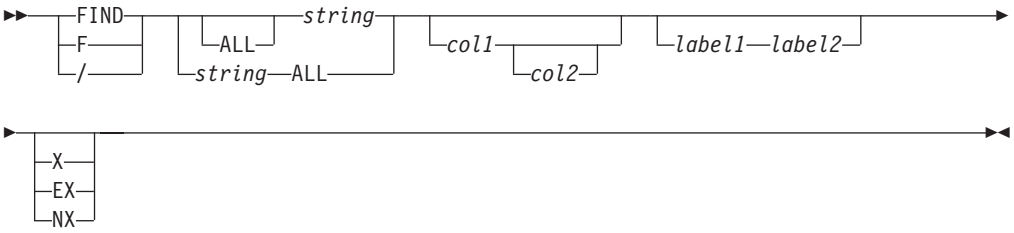
This primary command adds a key field to the current record. The command is available only when a single record is displayed using the ZOOM command.

FIND

This primary command finds the next occurrence of a character string in the data being browsed. When the string is found, File Manager positions the cursor at the beginning of the string and, if necessary, scrolls the data to bring the string into view.

There are two forms of the FIND command. The first form is simply FIND *string*. The second form, as detailed below, is only available when zoomed on a disk track.

Syntax



Parameters

ALL Searches forwards from the top of data. Same as FIRST, except that ALL also displays the total number of occurrences of *string* in the records searched.

Note: Suppressed or not-selected records that are hidden from display or represented by shadow lines are not searched, even when you specify ALL.

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length. If not specified, the last column of the record is used.

EX Excluded records only.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

NX Non-excluded records only.

X Same as EX.

string Search string. Can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, `Mixed` matches `MIXED`.
- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C).

The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.

- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).

To find the next occurrence of the same string, press the RFind function key (F5), or enter the FIND command with no parameters. A message is displayed if the string cannot be found.

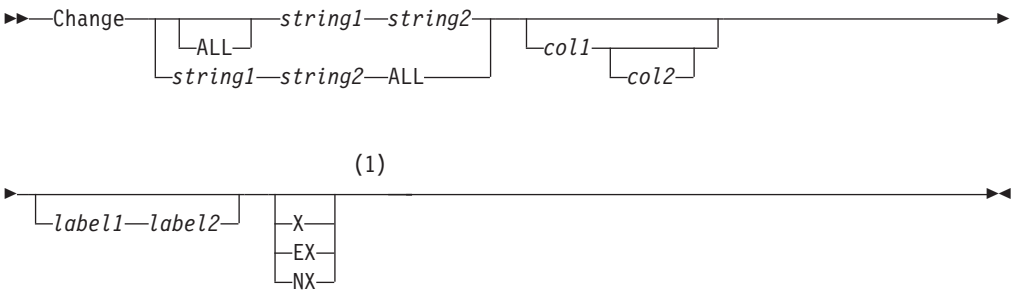
CHANGE

Use this primary command to locate and replace one or more occurrences of a character string in the data being edited.

The CHANGE command only affects the record you have zoomed in on.

After the data has been changed, File Manager places the cursor at the beginning of the changed string, automatically scrolling, if necessary, to bring the string into view.

Syntax



Notes:

1 If none of these parameters (X, EX, or NX) are specified, excluded and non-excluded records are searched.

Parameters

ALL Causes the search to begin at the top of the data and find and replace all occurrences of the string. If you do not limit the search to non-excluded records, then all excluded records containing the string are shown and replaced.

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length. If not specified, the last column of the record is used.

EX Excluded records only.

label1 Label identifying the start of a range of records. The label must start with a

period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

NX Non-excluded records only.

string1

The string you want to search for. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, the following command changes the strings `black`, `Black`, and `BLACK`:

  ```
  CHANGE black white
  ```

- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`. The string can be a null string (`''`). If *string1* is a null string, *string2* is inserted at the current column position.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`. The string can be a null string (`''`). If *string1* is a null string, *string2* is inserted at the current column position.
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).

string2

The string you want to replace *string1*. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. If CAPS ON or CASE UPPER is in effect, *string2* is translated to uppercase.
- A character string enclosed in quotation marks. It can contain blanks and commas. Case is respected and retained unless CASE UPPER is in effect, in which case *string2* is translated to uppercase. The string can be a null string (`''`).
- X followed by a hexadecimal string enclosed in quotation marks (`X'C1C2'`). If CASE UPPER is in effect, all hexadecimal representations of lowercase characters are translated to their equivalent uppercase characters.

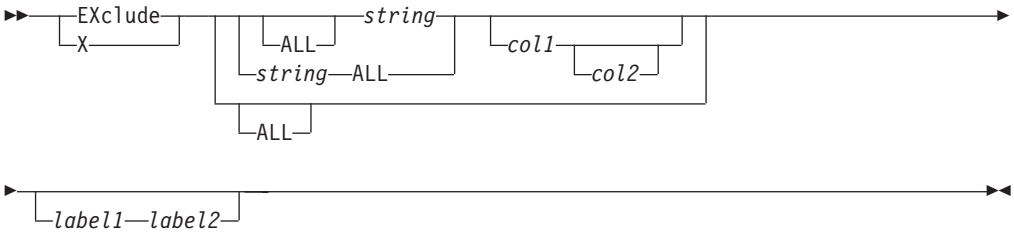
X Same as EX.

See "CHANGE/CX primary command" on page 748 for more information about the parameters used with the CHANGE primary command.

EXCLUDE

The EXCLUDE primary command excludes from display certain records in the data being edited.

Syntax



Parameters

ALL Causes the search to begin at the top of the data and find all occurrences of *string*. If not specified, the search begins at the cursor location (if the cursor is within the data portion of the screen) or the beginning of the first record displayed, and searches ahead to find the next occurrence of *string* in a record that is not already excluded.

Note:

1. The command EXCLUDE ALL excludes all displayed records.

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length. If not specified, the last column of the record is used.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

string The search string you want to search for. Records containing this string, within the limits imposed by the other EXCLUDE command parameters, are excluded. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, `Mixed` matches `MIXED`.
- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.

- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by X ('C1C2'X).

Disk Print (option 5.3)

Use this option to print the records in a disk extent in character or hexadecimal format.

Specifying what you want to print

To print records from a disk, you must specify the range of tracks that you want to print.

1. Specify the beginning of the range, in one of the following ways:
   - The beginning address in the form *cylhd*, where the last two digits represent the head and the other digits represent the cylinder
   - A track number, in the form R*nnnnnn*
   - An asterisk (*), which means the beginning of the data set
2. Specify the end of the range, in one of the following ways:
   - The end address in the form *cylhd*
   - A track number, in the form R*nnnnnn*
   - A number of tracks, in the form N*nnnnnn*
   - An asterisk (*), which means the end of the data set

Depending on the value of the **CYLHD** entry field of the Set Processing Options, you specify either the absolute disk location or the relative location within the data set.

If a defective track with an assigned alternate track is encountered, File Manager processes the contents of the alternate track.

When you print records from a disk, you can specify any of the following dump types:

KD Prints the key fields (if any) and data fields. This is the default.

CKD Prints the home address and record 0 fields, followed by the count, key and data fields.

KEY Prints *only* the key field and not the data portion of each record

CNT Prints *only* the home address and count fields and not the key field and data portion of each record

Controlling print output

Several options on the Set Processing Options (option 0) panel affect the printed output:

DATAHDR
Whether the output contains a data header for each record

DUMP
Which format (updown or across) is used for hexadecimal output

HEADERPG
Whether the output contains a header page at the beginning

PAGESIZE
How many lines per page the output has

PAGESKIP
Whether output from each function starts on a new page

PRINTLEN
How many columns wide the output is
PRINTOUT
Where the output is sent
PRTTRANS
Whether non-printable characters are translated to blanks
RECLIMIT
Which part of each record to print

For more information, see "Set Print Processing Options panel (option 0.1)" on page 658.

You can also specify a record length, to deblock each physical record into logical records. The physical block size does not need to be a multiple of the record length that you specify.

Disk Record Scan (option 5.4)

Use this option to find specific data on a disk.

Limit the search to a range of records by entering an extent begin and extent end, as you do for the print function. (See "Specifying what you want to print" on page 378 for more information.)

You must specify the type of scan you want performed by entering a value in the **Scan type** field. Values are:

D Searches the data field of each record.

K Searches the key field of each record.

E Searches for end-of-file records. If you select E, you do not need to specify a search string.

Scanning starts at the scan position (default 1). If you specify that pattern scanning is to be used, then scanning looks for the occurrence of the scan argument anywhere from the start position. If you specify that pattern scanning is not to be used, then scanning looks only at the start position.

If you specify a record length, physical blocks are deblocked into logical records before searching.

The scan argument is the string that you want File Manager to search for. The string can contain up to 50 characters. It has the same format as the find string (see "Disk Browse (option 5.1)" on page 371). The string does not apply if the scan type is E. Otherwise, it is required.

The output of this option is a list of records that meet your search conditions.

Write EOF Record (option 5.5)

Use this option to write a logical EOF marker into a sequential disk file.

File Manager retrieves the record at the specified location and rewrites it as an end-of-file indicator. File Manager writes a special record with a data length of zero (null record). All data on the track past this EOF record is lost. File Manager prints a log of the retrieved record, provided your print output is not routed to the

terminal. Alter the destination of the print log by changing the value in the **PRINTOUT** entry field of the Set Print Processing Options panel (see "Set Print Processing Options panel (option 0.1)" on page 658). The EOF record has a data length of zero.

Data Set Extents (option 5.6)

Use this option to display the begin and end of the extents of a data set.

The display panel shows absolute disk extents and disk extents relative to the beginning of the data set. You can use this information for all disk functions which prompt you for the location of a disk extent.

For multivolume data sets only the extents on the selected volume are shown.

Figure 108 shows an example of DSX output.

```
 Process   Options   Help
 File Manager                  Data Set Extents

Input:
                          File Manager Messages
 VOLSER: MVS4W4  Data Set Name: FMNUSER.FMOS390.QSAM

 Absolute and relative extent limits within data set:

                Begin - ABSOLUTE - End          Begin - RELATIVE - End
Ext  Tracks  Cyl-hd Rnnnnn   Cyl-hd Rnnnnn   Cyl-hd Rnnnnn   Cyl-hd Rnnnnn
  0      1    69  0   1035    69  0   1035     0  0      0     0  0      0
  1      1    69  1   1036    69  1   1036     0  1      1     0  1      1
  2      1    69  2   1037    69  2   1037     0  2      2     0  2      2
  3      1    69 13   1048    69 13   1048     0  3      3     0  3      3
  4      1    70  4   1054    70  4   1054     0  4      4     0  4      4

 Press the CANCEL or EXIT key to remove the message Pop-Up              6

 Command ===> ____________________________________________________ Scroll PAGE_
  F1=Help        F2=Split       F3=Exit        F4=CRetriev    F7=Backward
  F8=Forward     F9=Swap        F10=Actions    F12=Cancel

Command ===> ____________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 108. Data set extents display

VSAM Update (option 5.7)

Use this option to display VSAM data and update records or control intervals.

Be careful when updating control intervals. If you make incorrect changes to control information, the data set may become unusable.

To update a record:

1. Locate the record you want to update. As well as using standard ways of scrolling or finding, you can overtype the RBA, key and slot fields. By default keyed access is used for keyed data, and addressed access for non-keyed data. Specifying a RBA for a KSDS sets addressed sequence for subsequent scrolling.
2. Select the record with the Zoom function key (F2).
3. Change the data, using standard edit techniques.

4. Press the Exit function key (F3) to save the changed record, or Cancel function key (F12) to discard the changes.

File Manager prints a log of the changed record, provided your print output is not routed to the terminal. Alter the destination of the print log by changing the value in the **PRINTOUT** entry field of the Set Print Processing Options panel (see "Printing from File Manager" on page 299). You can select another record to update or leave the function.

To specify control interval access to see full control intervals (including control information) instead of logical records, enter YES in the **CI access** field. Control intervals are processed in addressed sequence. You cannot use CI access with a compressed KSDS (this is a system restriction).

VSAM update uses the same primary commands as "Disk Track Edit (option 5.2)" on page 372.

Data in Virtual (option 5.8)

Use this option to browse a data-in-virtual data set at the terminal.

Chapter 10. OAM Functions

Use the options of the OAM Functions menu to display, backup, and restore the contents of OAM collections or optical volumes. When you work with one object, you can browse, print, or update object data, and copy or erase objects.

These functions use the same primary commands as the Disk/VSAM Data Functions (option 5).

Introducing the Object Access Method

This section provides a brief introduction to Object Access Method (OAM). For more information, see the *z/OS DFSMS Object Access Method Application Programmer's Reference*.

In OAM, an *object* is a stream of bytes with a name. The content, format, and structure of that byte stream are unknown to OAM. For example, an object could be a compressed scanned image, or coded data. There are no individual records within an object.

A *collection* is a group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. Every object must be assigned to a collection. Object names must be unique within a collection; the same object name can be used in different collections.

OAM is an access method of MVS/DFP, the base for DFSMS. OAM uses the SMS-supplied hierarchy definition and management parameters to place user-accessible objects anywhere in the storage hierarchy. The hierarchy consists of DASD, library-resident optical volumes, and shelf-resident optical volumes. The location of an object in the hierarchy is unknown to the user; device-dependent information (such as track size) is not required.

For each object, the user can specify:

Storage Class
The level of service for the object, independent of the physical device or medium containing the object

Management Class
Backup, retention, and class transition characteristics for the object

The storage class and management class may be overridden by your installation defaults.

Object Directory List (option 6.1)

Use this option to list the objects from a collection, or a primary or backup optical volume. You can print the list, or, if you display the list, you can browse, print, update, or erase any of the listed objects.

Select the objects by a fully qualified or a generic object name, and specify the collection name, the volser of an optical volume, or both.

The information listed includes:

- Object name
- Object size
- Object location (DASD or optical)
- Volume serial number (if known)
- Creation and expiration dates
- Storage class
- Management class

Sort the list by object name, size, creation date, or expiration date.

You can use the display as a selection list. To invoke a File Manager command for an object, position the cursor on the desired line, press Process function key (F6), and select the function from the pull-down menu, or type a line command over the beginning of the object name.

The following figures show the object directory entry and display panels.

```
Process   View   Options   Help
_____________________________________________________________________________
File Manager           ODL - Object Directory List               Line 1 of 12

Collection FMNUSER.OAM.CLLCT01                          SC DB2DASD  MC MAGONLY

---- Object Name ---- sorted by NAME__ ----- ----Creation---- -Object- Location
....5...10...15...20...25...30...35...40.... ---Date--- -Time --size-- - VOLSER
FMNUSER.SIMOBJ.T01                          2000-03-09 16:48     7040 D   N/A
FMNUSER.SIMOBJ.T02                          2000-03-09 16:48     7865 D   N/A
FMNUSER.SIMOBJ.T03                          2000-03-09 16:48    13566 D   N/A
FMNUSER.SIMOBJ.T04                          2000-03-09 16:48    14280 D   N/A
FMNUSER.SIMOBJ.T05                          2000-03-09 16:49   242028 D   N/A
FMNUSER.SIMOBJ.T06                          2000-03-09 16:49     6776 D   N/A
FMNUSER.SIMOBJ.T07                          2000-03-09 16:49     1596 D   N/A
FMNUSER.SIMOBJ.T08                          2000-03-09 16:49   223380 D   N/A
FMNUSER.SIMOBJ.T09                          2000-03-09 16:49    17760 D   N/A
FMNUSER.SIMOBJ.T10                          2000-03-09 16:49     5445 D   N/A
FMNUSER.SIMOBJ.T11                          2000-03-09 16:49    11704 D   N/A
FMNUSER.SIMOBJ.T12                          2000-03-09 16:49   669696 D   N/A

Command ===> _____________________________________________________ Scroll PAGE_
F1=Help  F2=Browse  F3=Exit  F4=CRetrieve  F5=RFind  F6=Process  F7=Up
F8=Down  F9=Swap  F10=Actions  F11=RgtLeft  F12=Cancel
```

Figure 109. Object Directory List display panel

By pressing the Rgtleft function key (F11) you can switch between two display formats to see all the information you would get on a full-width printout. The second display format is shown in Figure 110 on page 385.

```
 Process   View   Options   Help
_____________________________________________________________________________
File Manager           ODL - Object Directory List                Line 1 of 12

Collection FMNUSER.OAM.CLLCT01                            SC DB2DASD  MC MAGONLY

---- Object Name ---- sorted by NAME__ ----- Storage  Managmnt Expiration Coll
....5...10...15...20...25...30...35...40.... --class- --class- ---date--- -ID-
FMNUSER.SIMOBJ.T01                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T02                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T03                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T04                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T05                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T06                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T07                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T08                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T09                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T10                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T11                           DB2DASD  MAGONLY  01.01.0001   11
FMNUSER.SIMOBJ.T12                           DB2DASD  MAGONLY  01.01.0001   11

Command ===> _____________________________________________________ Scroll PAGE_
F1=Help  F2=Browse  F3=Exit  F4=CRetrieve  F5=RFind  F6=Process  F7=Up
F8=Down  F9=Swap  F10=Actions  F11=RgtLeft  F12=Cancel
```

Figure 110. Object Directory List display panel (alternative display)

To move forward or backward through the library member list, use the Down function key (F8) or Up function key (F7), or use the LOCATE command.

Object Browse (option 6.2)

Use this function to browse an OAM object.

Browsing objects is similar to general browsing. For more information, see Chapter 3, “Viewing and changing data sets,” on page 49.

Object Print (option 6.3)

Use this option to print an OAM object in character or hexadecimal format.

You can specify the block size for the output, the number of bytes to skip from the beginning of the object, and the maximum number of bytes to print.

Controlling print output

Several options on the Set Processing Options (option 0) panel affect the printed output:

DATAHDR
: Whether the output contains a data header for each record

DUMP
: Which format (updown or across) is used for hexadecimal output

HEADERPG
: Whether the output contains a header page at the beginning

PAGESKIP
: Whether output from each function starts on a new page

PRINTLEN
: How many columns wide the output is

PRINTOUT
: Where the output is sent

PRTTRANS
: Whether non-printable characters are translated to blanks

RECLIMIT
Which part of each block to print

For more information, see "Set Print Processing Options panel (option 0.1)" on page 658.

Object Update (option 6.4)

Use this option to update an object.

To locate specific data, use the standard locate and positioning commands.

To update an OAM object:

- Overtype the existing text with new text.
- Enter a column number in the **Col** field, and use the GET or COPY command to overwrite data, starting at the specified column.

  For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns that you want to copy. For example, if the value in the **Col** field is 5 and you enter `COPY 10-12`, the data in columns 10 to 12 replaces the data that was in columns 5 to 7. If you enter `COPY 10`, the character in column 10 replaces the character that was in column 5.

Use the Exit function key (F3) to save your changes or Cancel function key (F12) to discard the changes.

Object Erase (option 6.5)

Use this option to erase an OAM object. When you specify the object, File Manager removes it from the collection.

Object Copy Functions (option 6.6)

When you select this option, you have a choice of copies. You can copy to and from VSAM, to and from QSAM, or object to object.

When you copy object to object, you copy to a different object collection, or to a different name within the same collection. You can optionally change the management class and storage class.

You can also use the copy options to back up and restore objects.

Backing up OAM objects

Use the To VSAM option to back up one or more OAM objects to a VSAM data set.

Use the To QSAM option to back up one or more OAM objects to a sequential data set.

Object to VSAM (option 6.6.1)

Use this option to copy an object to a VSAM data set, for backing up (as described below).

Object to Sequential Data (option 6.6.2)

Use this option to copy an object to a QSAM data set, for backing up (as described below).

VSAM to Object (6.6.3)

Use this option to copy an object from a VSAM data set, for restoring (as described below).

Sequential Data to Object (option 6.6.4)

Use this option to copy an object from a QSAM data set, for restoring (as described below).

Object to Object (option 6.6.5)

Use this option to copy an object within a collection, using a different object name, or into another collection.

Backing up a single OAM object

To back up a single OAM object, specify:

- The collection name and object name for the object.
- The output data set, according to the normal rules for sequential output or VSAM output functions.
- For VSAM output, whether you want to replace any existing records in the data set.
- Whether you want to create an object header. (An object header is a header record with directory information for every object copied, for later use in object restore operations.)

  If you want only the data within the object, specify that you do not want an object header. For VSAM, the output data set can be a linear data set, an ESDS or RRDS with an appropriate block size, or a KSDS (if the OAM object contains information that can be used as keys).

  If you want the object as such (for example, to create a backup copy that you might later copy back to an OAM object collection), you should specify that you want an object header. For VSAM, you should use an ESDS.

  The object header is intended for use by the File Manager option only. It is not designed for use by products other than File Manager.
- The block size for the output data set. The default is 4096.
- The other input parameters (such as *limit* parameter or the optical disk volume serial number) apply when you are backing up a single object.

Backing up multiple OAM objects

To back up multiple OAM objects, specify:

- Which objects you want to back up. You must specify a collection name, an optical disk volume serial number, or both. You can back up all of the objects within the collection, volume, or collection/volume combination, or you can back up a subset of the objects based on these criteria:
  - A generic object name, containing at least one percent sign (%), asterisk (*), or double asterisk (**). (With a non-generic object name, you back up a single object as described previously.)

  - The *limit* parameter, to limit the backup to objects on DASD, objects on optical disks, or objects on a specific optical disk volume. (If you specify an optical disk volume serial number, you must also specify VOLSER for the *limit* parameter.)
  - The earliest creation date
  - The latest creation date
- The output data set, according to the normal rules for sequential output or VSAM output functions. For VSAM output, the data set must be an ESDS.
- For VSAM output, whether you want to replace any existing records in the data set.
- The block size for the output data set. The default is 4096.
- The *header* parameter does not apply when you are backing up multiple objects. A header is automatically generated for each object. This object header is intended for use by File Manager options only. It is not designed for use by products other than File Manager.

Restoring OAM objects

Use the From VSAM option to restore one or more OAM objects from a VSAM data set.

Use the From QSAM option to restore one or more OAM objects from a sequential data set.

To restore one or more objects, you should know whether the backup data set was created by a single-object backup or a multiple-object backup.

Restoring an object that was backed up individually

The instructions in this section tell you how to:
- Restore an object that was backed up as a single object with an object header.
- Restore an object that was backed up as a single object without an object header.
- Create an OAM object from a data set that was not previously an OAM object. (This is equivalent to the previous case, because File Manager does not know if a data set with no header was formerly an OAM object.)

In any of these cases, specify:
- The input data set, according to the normal rules for VSAM input or sequential input functions.
- The collection name, object name, storage class, and management class. By default, the values in the object header (if any) are used.
- Whether you want to replace any existing object with the same name in the same collection.

Restoring one or more objects from a multiple-object backup

Follow the instructions in this section to:
- Restore all of the objects from a multiple-object backup.
- Restore some of the objects from a multiple-object backup.
- Restore one of the objects from a multiple-object backup.

In any of these cases, specify:
- The input data set, according to the normal rules for sequential input or VSAM input functions.

- Which object or objects you want to restore. You can restore all of the objects in the data set, or you can restore a subset of the objects based on these criteria:
  - Collection name
  - Object name or generic object name
  - The *limit* parameter, to restore only objects that were backed up from DASD, from optical disks, or from a specific optical disk volume.
  - The earliest creation date
  - The latest creation date
- The storage class and management class.
- Whether you want to replace any existing object with the same name in the same collection.

The object or objects that meet the specified criteria are restored. Each object has the same object name and the same collection as it had before it was backed up. You cannot change the object name or collection name with this function.

Chapter 11. Using UNIX System Services and the Hierarchical File System

z/OS UNIX System Services (USS) gives the z/OS operating system an open standards interface. It contains a UNIX shell and utilities, which you can use to enter shell commands, write shell scripts, and work with the file system.

z/OS UNIX provides a Hierarchical File System (HFS) for z/OS. A file within z/OS UNIX is called an HFS file. HFS files are organized in a hierarchy of files and directories in a tree much like UNIX. A directory can contain files or other subdirectories.

To access the Hierarchical File System:

1. Select option 8 (**HFS**) from the Primary Option Menu.

   File Manager displays the Access Hierarchical File System panel.
2. On the Access Hierarchical File System panel, select the required option so that File Manager invokes standard ISPF services to enable access to USS and HFS utilities.

If you are familiar with File Manager functions, there is a convenient way to access HFS files. For basic functions, you can specify an HFS file in the same panels and batch commands as an z/OS data set.

File Manager can access an HFS file as a simulated QSAM/BSAM file. This means that at the I/O level, the file is seen as a single-volume, physical, sequential data set that resides on DASD. It is allocated dynamically or referred using a DD statement (PATH parameter).

The following sections summarize the main features and differences.

Specifying an HFS file

An HFS file is identified by means of a path leading to it. File Manager requires an absolute path; file specification starts from the root. Paths are specified in the same fields and using the same batch parameters as z/OS data sets. A name is considered a path name if contains a slash ("/"). A correct absolute path name must start with a slash, otherwise, it is treated as a data set name.

A path name is case-sensitive and can contain spaces, commas, and other special characters. In batch, the path name needs to be enclosed in single quotes. In fullscreen mode, use of apostrophes is optional.

File Manager allows for an absolute path name up to 255 characters long. Fields in panels dedicated for data set names can also carry path names. They are expandable and allow for long name specification. Also, the syntax used in batch allows for specification of path names, even if they do not fit into one line.

In batch, you can specify the path name with a PATH JCL parameter, and refer to it in File Manager control statements with a ddname. For a description of PATH and other path-related parameters, see the JCL manuals.

The following File Manager functions support HFS files:

- Edit
- View
- Create (DSG)
- Copy (DSC)
- Print (DSP)
- Compare (DSM)
- DSEB (in Browse mode only)
- Find (FCH)

Using directories

HFS files are maintained in directories, which constitute a hierarchical structure, not supported by standard z/OS data management access methods. However, File Manager allows for specifying a path to a directory (like specifying a PDS library without member).

An HFS directory is displayed in a similar way to a PDS directory. Each entry on the list contains an object name and basic attributes. An entry in the directory can describe another directory. Navigations through the nested lists of entries is supported. The object name may not fit into one line of the list, with only a part of name being displayed. To see the whole name, position the cursor at the name and press Enter to display a pop-up window showing the absolute path name. For the copy operation, the pop-up window also allows you to specify the target file or a PDS member name (if copying to a PDS library).

Selecting files for processing

An absolute path name may specify a file, a group of files with names matching a pattern, or a directory. In on-line mode, the scope of files displayed or processed depends on the last component of the path name specified as shown in Table 9.

Table 9. Scope of files displayed or processed according to pathname specified

Last component of path name specified	**Files displayed or processed**
A file	The file is processed.
A directory	A list of all names in the directory is displayed. You can select files to be processed.
A name containing wildcard characters	A list of all matching names is displayed. You can select files to be processed.
An asterisk (*)	Where possible (for example, Print, Copy, Find), all files in the directory (but not in subdirectories) are processed without displaying the list of names.

When you are creating JCL and control statements for submission (in batch), the rules are the same. However, you can only select one file for further processing.

In batch, as it is not possible to navigate through subdirectories and dynamically select files, all files which are referred to by a path name are processed. In the other words, the rules are the same as when processing on-line, except that all files within the scope determined by the path name are selected.

Logical record length

HFS files are byte-oriented, not record-oriented like z/OS data sets. An HFS file is either defined as a *text* file, where record boundaries are determined by a delimiter (such as NL, LF, CR, and so on), a *binary* file (no specific boundaries), or neither a text or binary file. When accessing the file as a simulated QSAM data set, File Manager must identify logical records. The choice is left to you, rather than based on the file type. File Manager assumes text mode (delimited records), unless you request binary mode. In text mode, records are variable-length with the data residing between delimiters. In binary mode, delimiters are not distinguished; the records are user-defined and of fixed length (80 is the default).

File Manager assumes a maximum block size of 32760 bytes. The maximum record length is 32752 for variable-length records, and 32760 for fixed-length records. If the record length is greater than the limit, you must use binary mode.

Processing an HFS file

File Manager allows for various processing at record level. A record simulated in an HFS file can be subject to the same processing as a record belonging to an z/OS data set. The similarity at record level enable HFS files to be viewed, edited, copied, and compared using the same functions and panels as z/OS data sets. Also, the same rules for using templates and manipulating data apply. Some specific differences are mentioned in the following sections.

Viewing and editing HFS files

HFS files are viewed and edited in the same way as z/OS data sets (see Chapter 3, "Viewing and changing data sets," on page 49). You can specify an existing HFS file or directory as input. When you use Edit, File Manager uses the in-memory technique if the file fits into memory, otherwise it uses an auxiliary edit. In-place edit is not used.

A viewed or edited file can be interpreted as a text or binary file. If the **Binary mode** option is not selected, text mode is assumed and records are determined by delimiters. If you select the **Binary mode** option, the records are determined by record size defined in reclen, or by the default, 80 bytes. In View or Edit, records in text mode are assumed to be variable-length and up to 32752 characters long. However, if the size of the whole file is less than the record size limit, file size is assumed as the maximum record size.

If you specify a path to the HFS directory as an input file, File Manager displays the directory list. You can select one or more HFS files and then view or edit them. If you select a subdirectory, File Manager displays a new list, allowing selection on the next level. To return to the previous list, press the Exit function key (F3). In this way, you can navigate through the directories and choose the files to be displayed.

To display a pop-up window with the absolute path name (including the complete file name, which may be shortened on the list), position the cursor at the file name and press Enter.

Creating HFS files

You can create new records in an existing HFS file in the same way as you do for z/OS files, or create a new file and initialize it with new records. The rules are described in "Creating data sets and records" on page 242. If the file already exists, specify the same options as for an z/OS file. If you are creating a new file, select **9 (HFS)** on the Allocate: New Data Set Organization panel. Then, on the File Attributes panel, define the privileges details for accessing the file.

A viewed or edited file can be interpreted as a text or binary file. If you do not select the **Binary mode** option, text mode is assumed and records are determined by delimiters. If you select the **Binary mode** option, the records are determined by record size defined in reclen, or by the default, 80 bytes.

You cannot specify a path to the HFS directory. Only a regular file can be allocated or initialized.

Copying HFS files

You can copy HFS files to and from other HFS files and z/OS data sets. In general, the methods are the same as described in "Copying data sets" on page 253. When you specify an HFS file as either the "From" or "To" object, File Manager treats it as a sequential data set.

Both source and target files can be interpreted as text or binary files. If the **Binary mode** option is not selected, text mode is assumed and records are determined by delimiters. If you select the **Binary mode** option, the records are determined by record size defined in reclen, or by the default, 80 bytes.

If you specify a path to the HFS directory as an input file, File Manager displays the directory list. You can select one or more HFS files and copy them. If you select a subdirectory, File Manager displays a new list, allowing selection on the next level. To return to the previous list, press the Exit function key (F3). In this way, you can navigate through the directories and choose the files to be displayed.

To display a pop-up window with the absolute path name (including the complete file name, which may be shortened on the list), position the cursor against the file name and press Enter. If this is the "From" HFS directory, the pop-up window allows you to rename the target file (if the "To" specification describes an HFS directory) or the target member (if the "To" specification describes a PDS library).

If the directory is specified in batch, all regular files from this directory, but not from sub-directories, are copied.

When you copy members from a PDS library to an HFS directory, member names become file names (they can be changed in the fullscreen mode). When you copy from an HFS directory to a PDS library, the targets are members and their names must conform to a member name. If you do not rename the target files, File Manager derives the file name in the following way:

- Translates the file name to uppercase.
- If the name is longer than 8 characters, characters from the ninth position onwards are ignored.
- Starting from the first character, File Manager uses the maximum number of characters (up to a maximum of 8) that conform to member name rules to form

the file name. If the first character violates the member name rules, the target name is not created and the file is not copied.

Comparing HFS files

An HFS file can be compared to another HFS file or to an z/OS data set. You can specify the "Old" or the "New" object (or both) as a path describing the file. The compare operation and options for an HFS file are the same as for a BSAM/QSAM data set and are as described in "Comparing data sets" on page 281.

A compared file can be interpreted as text or binary files. If the **Binary mode** option is not selected, text mode is assumed and records are determined by delimiters. If you select the **Binary mode** option, the records are determined by record size defined in reclen, or by the default, 80 bytes.

If you specify a path to an HFS directory as the "Old" or "New" file, the directory list is displayed (fullscreen mode). You ust select an HFS file from the list before the compare starts. The subjects of the compare are always files and data sets, not directories.

To display a pop-up window with the absolute path name (including the complete file name, which may be shortened on the list), position the cursor against the file name and press Enter.

You cannot specify a directory in batch.

The compare output does not differ significantly from the output produced when data sets are compared. The only difference is "file" is used instead of "data set".

Printing HFS files

You can print HFS files in the same way that you print z/OS files and as described in "Printing from File Manager" on page 299.

A printed file can be interpreted as text or binary files. If the **Binary mode** option is not selected, text mode is assumed and records are determined by delimiters. If you select the **Binary mode** option, the records are determined by record size defined in reclen, or by the default, 80 bytes.

If you specify a path to the HFS directory as the input file, File Manager displays the directory list. You can select one or more HFS files for printing. If you select a subdirectory, File Manager displays a new list, allowing selection on the next level. To return to the previous list, press the Exit function key (F3). In this way, you can navigate through the directories and choose the files to be displayed.

To display a pop-up window with the absolute path name (including the complete file name, which may be shortened on the list), position the cursor against the file name and pressing Enter. If the directory is specified in batch, all regular files from this directory, but not from sub-directories, are printed.

The printed output does not differ significantly from the output produced when you print data sets. The only difference is "file" used instead of "data set".

Finding a string in HFS files

You can search HFS files in the same way that you find strings in z/OS files as described in “Finding and changing data in multiple PDS members” on page 270.

A searched file can be interpreted as text or binary. If the Binary mode option is not selected, text mode is assumed and records are determined by delimiters. If you select the Binary mode option, the records are determined by record size defined in reclen, or by the default, 80 bytes.

If you specify a path to the HFS directory as the input file, File Manager displays the directory list. You can select one or more HFS files for processing. If you select a subdirectory, File Manager displays a new list, allowing selection on the next level. To return to the previous list, press the Exit function key (F3). In this way, you can navigate through the directories and choose the files to be processed. This allows you to build a process list which you can further refine and search.

To display a pop-up window with the absolute path name (including the complete file name which may be shortened on the list), position the cursor against the file name and press Enter.

If the directory is specified in batch, all regular files from this directory, but not from sub-directories, are searched.

The output from the find process does not differ significantly from the output produced when you search data sets. The only difference is the words "path" and "file" are used instead of "data set" and "member" respectively.

Related topics

“Primary Option Menu panel” on page 615
“Access Hierarchical File System panel” on page 438

Chapter 12. Introduction to programming with File Manager functions

Most of the tasks that you can perform using File Manager panels under ISPF can also be performed by calling equivalent File Manager functions from batch jobs, REXX procedures or TSO clists.

Summary of File Manager functions

File Manager functions are listed below, arranged according to the tasks they can perform, and the data types you can use them with.

Where a File Manager function has an equivalent File Manager panel that you can use under ISPF, the panel option number is listed under the "Equivalent panel" column.

For a list of functions where you can use templates, see "Where can you use templates?" on page 7.

Functions marked "(batch only)" cannot be used in REXX procedures or TSO clists.

Listing catalogs and directories

What you can list	Function	Equivalent panel
System catalog entries	SCS	3.4
VSAM catalog entries	SCS	3.4
VTOC entries	DVT	3.5
OAM objects	ODL	6.1
MQ Managers	MQL	9.1
MQ Queues	Queues	9.1 after selecting a manager

Editing, updating, comparing and creating data

What you can edit	Function	Equivalent panel
VSAM data	DSEB (batch only)	2
Sequential data	DSEB (batch only)	2
CICS resource	DSEB (batch only)	2

What you can update	Function	Equivalent panel
VSAM data	DSU (batch only)	
Sequential data	DSU (batch only)	
CICS resource	DSU (batch only)	

Note: The difference between using DSU to "update" data versus using DSEB to "edit" data is that DSU operates on a "per record" basis: each record in a data set

is treated separately, in sequence, whereas you can use DSEB to move between the records in a data set as you choose, similar to the way you can scroll around a data set in an edit panel.

What you can compare	Function	Equivalent panel
VSAM data	DSM	3.11
Sequential data	DSM	3.11
MQ data	DSM	3.11
CICS resource	DSM	3.11

What you can create	Function	Equivalent panel
Sequential data	DSG	3.1
VSAM data	DSG	3.1
MQ data	DSG	3.1
Tape data	BT	4.7
CICS resource	DSG	3.1

Copying data

Copy from	Copy to (function, equivalent panel)						
	VSAM	**QSAM**	**Tape**	**OAM**	**REXX**	**MQ**	**CICS**
VSAM	DSC, 3.3	DSC, 3.3	VT, 4.2.6	VO, 6.6.3	VX	DSC, 3.3	DSC, 3.3
QSAM	DSC, 3.3	DSC, 3.3	ST, 4.2.7 QT, 4.2.7	SO, 6.6.4 QO, 6.6.4		DSC, 3.3	DSC, 3.3
Tape	TV, 4.2.4	TS, 4.2.5 TQ, 4.2.5	TT, 4.2.1 TTR, 4.2.2 TLT, 4.2.3		TX	DSC, 3.3	DSC, 3.3
OAM	OV, 6.6.1	OS, 6.6.2 OQ, 6.6.2		OO, 6.6.5			
REXX	XV		XT				
MQ	DSC, 3.3	DSC, 3.3	DSC, 3.3			DSC, 3.3	DSC, 3.3
CICS	DSC, 3.3	DSC, 3.3	DSC, 3.3			DSC, 3.3	DSC, 3.3

Printing data

What you can print	Function	Equivalent panel
VSAM data	DSP	3.2
Sequential data	DSP	3.2
MQ data	DSP	3.2
CICS resource	DSP	3.2
Tape data	TP	4.5
Tape data	TP	4.5
CKD disk tracks	DP	5.3
OAM objects	OP	6.3

Erasing data

What you can erase	Function	Equivalent panel
Tape records	ERT	4.13
OAM objects	OE	6.5

Managing data sets

What you can do	Function	Equivalent panel
Alter a data set	DSFA	
Allocate a data set	DSFC	
Delete a data set	DSFD	
Rename a data set	DSFR	

Managing tapes

What you can do	Function	Equivalent panel
Space backward by record	BSR	4.15.3
Space backward by file	BSF	4.15.1
Space forward by record	FSR	4.15.4
Space forward by file	FSF	4.15.2
Rewind	REW	4.15.5
Rewind and unload	RUN	4.15.6
Initialize	INT	4.12
Write tape marks	WTM	4.11
Compare two tapes	TTC	4.9
Print tape label information	TLB	4.8
Print a tape map	TMP	4.6

Managing your processing environment

What you can do	Function	Equivalent panel
Query File Manager version	VER	See note
Format DBCS data	FMT	3.0
End File Manager job	EOJ	
Change processing options	SET	0

Note: To query the File Manager version when running File Manager under ISPF, open the **Help** pull-down menu, then select **About**.

Using File Manager functions in REXX procedures or TSO clists

To call a File Manager function from a REXX procedure, use the following syntax:

```
FILEMGR "$function-name-parameters"
```

To call a File Manager function from a CLIST, use the following syntax:

```
FILEMGR $function-name-parameters
```

Join the parameter names and values with an equal sign or a blank. Separate these pairs with a comma or blank.

For example:

```
FILEMGR "$DSP DSNIN='FMNUSER.TEST.KSDS1',FORMAT=HEX,KEY=000100,NLRECS=20"
```

Within a REXX procedure, all profile settings are taken from the installation defaults. For details on changing File Manager default profile settings, see *File Manager Customization Guide*.

Example

The following example shows a REXX procedure that uses File Manager functions.

```
/* REXX */
/* ************************************************/
/* Example: Execute File Manager functions from REXX */
/* ************************************************/
parse upper source environment.
/* List the catalog entries on the printout                      */
FILEMGR "$SCS FUNCTION=PRINT"   /* call SCS for the master catalog */
say "SCS return code:" RC
/* Provided the installation default for PRINTOUT is PRINTOUT=REXX */
/* then commands from this platform will be directed to the stem   */
/* variable FILEM.                                               */
filem.0=0                       /* discard any previous output     */
                                /* call SCS for a user catalog     */
FILEMGR "$SCS CATALOG='CATALOG.PRODUCTS.UCAT',DSNAME='ADSM.**'," ,
      "FUNCTION=PRINT,SORTBY=DATE"
say "SCS return code:" RC
/* Display the FILEM.n variable contents                         */
SAY ">>> SCS output:"
DO i=1 to filem.0               /* process all printed lines       */
  SAY filem.i                   /* display the printline           */
  END
```

Figure 111. Example REXX procedure using File Manager functions

Invoking File Manager panels from REXX procedures or TSO clists

You can also directly invoke File Manager panels from any REXX procedure or TSO clist (or ISPF Command line). For example, from any ISPF Command line, you can enter the following command to display the File Manager Print Utility (option 3.2) panel:

```
FILEMGR DSP
```

(that is, as if you were calling the function, but with no $ before the function name and no parameters following it, except for DSB, DSE, and DSV as described below)

Note: Except for View (option 1) and Edit (option 2), you cannot specify any parameters when you use this method to invoke a panel. If you specify a valid data set name after DSB, DSE, or DSV then you bypass the entry panel (where you would normally specify the data set name), and go directly to viewing or editing the specified data set.

For a complete list of the equivalent function names you can use to invoke File Manager panels, see “Summary of File Manager panels” on page 433.

Invoking File Manager functions from a REXX procedure or CLIST in /BATCH mode

Some functions which do not require user interaction (such as DSC, DSP, DVT, FCH, and SCS) and are otherwise available for JCL batch processing may also be invoked from a REXX program under TSO/ISPF in a "pseudo-batch" mode. This can be achieved by specifying a /BATCH parameter on File Manager invocation. When specified, it forces batch-type processing while executing under TSO/ISPF. Prior to program invocation, you must allocate all the required data sets, including SYSIN and SYSPRINT. All program output is written to the data set defined by SYSPRINT. The following shows a sample program invocation from a REXX program using the /BATCH parameter:

```
/* rexx */
"ALLOC DD(SYSPRINT) NEW LRECL(132) RECFM(F B) DSORG(PS) REUSE SPACE(1 1) CYL"
"ALLOC DD(DDIN) DA('TEST.PDS') SHR REU"
"ALLOC DD(SYSIN) NEW LRECL(80) RECFM(F B) DSORG(PS) REUSE SPACE(1) TRA"
PARM.0 = 3;
PARM.1 = "$$FILEM SET HEADERPG=NO"
PARM.2 = "$$FILEM FCH MEMBER=* INPUT=DDIN "
PARM.3 = "C '12AFSDF' '2ASDFSD'"
"EXECIO * DISKW SYSIN (STEM PARM. FINIS"
"CALL *(FMNMAIN) '/BATCH'"
/* or "FILEMGR '/BATCH'" */
SAY RC
"EXECIO * DISKR SYSPRINT (STEM SP. FINIS"
DO I = 1 TO SP.0
SAY SP.I
END
"FREE DD(DDIN SYSIN SYSPRINT)"
```

Using File Manager functions in batch jobs

To use a File Manager function in a batch job, include the following EXEC statement:

```
// EXEC PGM=FILEMGR
```

You must supply a DD statement for SYSPRINT. You might also need a STEPLIB statement or other DD statements, depending on which File Manager functions the batch job contains.

When you code File Manager control statements, you must:

- Begin each control statement with `$$FILEM` in columns 1 to 7, followed by a blank in column 8, followed by the command name, followed by the first parameter (if any).
- Specify parameters and their values separated by an equal sign, like this:
  `PARAMETER=`*`value`*
- Follow each parameter (except the last one) immediately with a comma.

Comment statements, which begin with an asterisk (*), are also allowed.

To continue a control statement onto the next line, you need a continuation character; simply end the first line with a comma and begin the next line with `$$FILEM` and a blank.

Similarly, if a parameter does not fit into one line, you do not need a continuation character. Simply split the parameter into more than one line, enclosing each part into apostrophes. for example:

```
$$FILEM DSNIN='u/test/this is a regular HFS file ',
$$FILEM 'which has v',
$$FILEM 'ery long name'
```

Note:

1. If the control statement contains only the function name, there must be a space between the function name and the comma.
2. To continue FIND and CHANGE commands in the FCH (Find/Change) utility, just end the first line with a comma (the continuation line does not contain $$FILEM).

Example

In HEX format, the following example shows a batch job that uses the DSP File Manager function to print 20 records of a data set (starting at the record with key value 0001000):

```
//jobcard
//ABC EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
$$FILEM DSP DSNIN='FMNUSER.TEST.KSDS1',
$$FILEM FORMAT=HEX,KEY=000100,NLRECS=20
/*
```

Setting processing options in batch

In batch, to change one or more options, or reset options to the installation default, use the SET function.

Related topics
"SET (Set Processing Options)" on page 1047

Destination of File Manager print output

Any File Manager print output that you produce in a batch job is always sent to SYSPRINT.

In a REXX procedure, the destination for File Manager print output depends on the value of the PRINTOUT parameter of the SET function:

SYSPRINT
Output is sent to the current SYSPRINT allocation.

SYSOUT=*c*
Output is accumulated in the data set specified by the PRINTDSN parameter of the SET function.

To browse your print output online, use Print Browse (option 3.9). This option is a shortcut to browsing the data set specified by the PRINTDSN parameter. To go to the Print Browse panel directly from any other File Manager panel, enter `PB` on the Command line. For information about using the Print Browse panel, see "Printing from File Manager" on page 299.

TERMINAL
Output is sent to the terminal.

REXX Output is written to a REXX stem variable. Each line of output corresponds to a variable named FILEM.*nnn*. FILEM.0 contains a line count.

After your first printing operation, lines of output are written in variables FILEM.1 to FILEM.*n*, and FILEM.0 has the value *n*. The next printing operation begins with FILEM.*n+1*.

Handling errors

File Manager sets the following return codes:

0 All File Manager functions processed successfully.

16 File Manager terminated because of an error. File Manager or system messages identify the error. For more information about File Manager messages, see "Messages" on page 1167.

>100 Not enough storage to load File Manager.

If an error is detected, File Manager issues an error message and terminates. If an error causes the system to cancel a File Manager function (for example, a file open error), File Manager tries to recover from the cancel condition and close any files left open before it terminates.

Some File Manager functions have their own return codes. If you receive a return code that is not listed above, refer to the description of the function in Chapter 16, "Functions," on page 831.

Note:

1. In addition to the return codes provided by File Manager, you can create your own custom return codes during the installation and customization of the product. These customized codes take precedence over any default File Manager return codes. The functions that can have customized return codes are:
   - CLM (Load Module Compare)
   - DSC (Data Set Copy)
   - DSEB (Data Set Edit Batch)
   - DSG (Data Set Generate)
   - DSM (Data Set Compare)
   - DSP (Data Set Print)
   - DSU (Data Set Update)
   - DVT (Display VTOC)
   - FCH (Find/Change)

   If you receive return codes that do not match the ones described here or in the Function Reference chapter, they might have been customized for your site. Please refer to your File Manager systems administrator for information.
2. In batch, you may want to transform an original or customized non-zero return code to an abend (Abend 999, Reason Code=888 (hex: 378)). It could be also forced by installation at your site. Abnormal termination of a File Manager step prevents execution of successors (steps and/or jobs). Please check the ABENDCC option of the SET command, or contact your File Manager system administrator for details.

Chapter 13. Enhancing File Manager processing

You can enhance normal processing of a number of File Manager tasks by supplying a procedure that runs in conjunction with the File Manager process. The procedure can comprise DFSORT statements, REXX statements, or a combination of DFSORT and REXX statements. There is a File Manager subset of the REXX language, called FASTREXX, that File Manager processes internally with high performance.

Why enhance File Manager processing?

The main benefit of using File Manager enhanced processing is that File Manager performs all data set input and output (I/O) operations for you, allowing you to concentrate on what you want to do with each data record. You can write record-handling instructions that are independent of the data file organization, without the need to write "peripheral" code for such things as I/O and error handling. This can greatly improve your productivity. Typically, when you use enhanced processing, you write small, easy-to-maintain "programs" that allow you to focus on the business task rather than on data processing tasks. You simply choose which File Manager utility to enhance, and then specify the processing that is desired for each record or library member.

In short, enhanced processing allows end users to easily perform tasks which would otherwise need to be done by skilled application programmers, and to concentrate on business results rather than data processing problems.

The tasks you can enhance

In the following situations, you can supply a procedure to enhance normal File Manager processing according to your own custom requirements:

Table 10. Situations in which you can supply a procedure with DFSORT or REXX statements

When performing this task...	REXX	DFSORT
Copying data with the Copy Utility (option 3.3) or DSC function	✓	✓[1]
Editing a data set with the DSEB function (not available in panels)[2]	✓	
Printing data with the Print Utility (option 3.2) or DSP function	✓	✓[1]
Updating records in a data set with the DSU function (not available in panels)[2]	✓	
Finding or changing data with the Find/Change Utility (option 3.6) or FCH function	✓	

Note:

1. DFSORT cannot be used on PDS(E)s.
2. You *must* supply REXX statements with the DSEB and DSU functions. In other situations, this is optional.

Choosing the right function to enhance

Sometimes, the ability to enhance File Manager functions and panels can make it unclear which one to invoke for a particular task, and what combination of template, DFSORT and REXX (or FASTREXX) processing is best to use. Typically, you get the best performance by using the function or panel that requires the least enhancement to perform the task, and by using the best-performing enhancement technique that can provide the desired result.

For more information about performance considerations, see “Choosing between FASTREXX, REXX, and DFSORT for your procedure” on page 408.

DSC function or Copy Utility (option 3.3)

First, consider whether you can perform the desired copying task by enhancing the function or panel with FASTREXX or DFSORT statements in a procedure. A task performed with only FASTREXX or DFSORT uses far less system resources than the equivalent task performed with REXX or template processing that requires REXX.

If neither FASTREXX or DFSORT are suitable for the copying task, before resorting to using REXX, consider whether the task can be performed with template processing that does not require REXX. You can also use template processing for selecting, reformatting and initializing records, and it uses less system resources than the equivalent task performed with REXX in a procedure. If you are using templates for record selection, try to code selection expressions that exploit the power of File Manager's internal expression processor, avoiding the need to invoke REXX to process the expressions (see “About expression processing” on page 213).

DSP function or Print Utility (option 3.2)

As with copying, you can use FASTREXX or DFSORT procedures or templates to help perform complex printing tasks without the overhead of external REXX processing. Using FASTREXX, DFSORT, or a template, you can specify complex criteria to select only certain records, and which fields from those records to print.

In some cases, you might want to use DSC or the Copy Utility (option 3.3) to reformat the fields in a data set and copy the reformatted fields to a new data set, before using DSP or the Print Utility (option 3.2) to print the new data set.

DSEB function (not available in panels)

DSEB is the only function where you can move forwards and backwards through the records in a data set, rather than dealing with each record independently, in sequence. Use DSEB when you need to compare values in different records, or where you need the flexibility to scroll or search forwards or backwards through a data set. Note, however, that DSEB only lets you “edit in place”. That is, you cannot insert or delete a record, or change a record's length.

DSEB has no default action of its own, such as printing or copying; you must supply a procedure containing REXX statements to perform whatever actions you require.

Before you decide to use DSEB, consider whether you can use DSU. DSU is faster than DSEB, especially with FASTREXX.

Note: FASTREXX processing is not supported for the DSEB function.

DSU function (not available in panels)
DSU is a general-purpose function for working with each record in a data set independently, in sequence. DSU is the same as the FCH function or the Find/Change Utility (option 3.6), except that DSU does not produce a report. Use DSU when you want to examine and potentially update each record, without needing to move backwards through the records or compare values of different records. However, before you decide to use DSU, consider whether you can use a combination of the primary commands available with the FCH function and the Find/Change Utility (option 3.6).

DSU only lets you "edit in place". That is, you cannot insert or delete a record, or change a record's length.

DSU has no default action of its own, such as printing or copying; you must supply a procedure containing REXX statements to perform whatever actions you require.

FCH function or Find/Change Utility (option 3.6)
When using this function or panel, you can either supply a procedure containing REXX statements (as described in this chapter), or a sequence of primary commands (such as FIND and CHANGE; for details, see "Finding and changing data in multiple PDS members" on page 270 and "FCH (Find/Change)" on page 1015). For better performance, use the primary commands instead of a REXX procedure wherever possible.

Creating and supplying a procedure

Your procedure, containing DFSORT or REXX statements or a combination of all of these, can be supplied as:

- A temporary procedure, created as you process a relevant File Manager panel, or invoke a relevant File Manager function from a REXX program;
- A saved procedure, nominated as you process a relevant File Manager panel or invoke a relevant File Manager function in batch or from a REXX program; or
- An inline procedure, nested in a batch job stream, that invokes a relevant File Manager function.

Saved procedures must reside in a member of a fixed or variable format PDS. The record length of the data set is unrestricted, but must be at least large enough for complete tokens to be entered, with consideration given to the statement continuation requirements of DFSORT (as processed by File Manager, see below) and REXX. A record length of 80 is suitable because it meets these requirements comfortably. With this length procedure statements can be copied into a temporary Edit session without truncation, or used as an inline procedure in a JCL Edit session.

Whether a procedure is entered at the terminal, saved in a PDS, or supplied as an inline procedure in JCL, it must follow the rules described below for File Manager procedures, as well as the specific rules for DFSORT and REXX statements defined by the DFSORT and REXX products.

Procedure statement types

File Manager supports two types of procedure statements, that you can use to enhance the File Manager functions. These are:

- **A subset of DFSORT statements**

DFSORT is IBM's high-performance sort, merge, copy, analysis and reporting product. DFSORT is an optional feature of z/OS, but File Manager can exploit its high-performance copy option for the DSC and DSP functions if certain installation requirements are met (for more information see the *File Manager Customization Guide*), and if the data to be processed is suitable. When these conditions are met, you can enhance the File Manager print and copy actions by supplying a procedure containing statements from the subset of the applicable DFSORT statements supported by File Manager, instead of using equivalent REXX statements.

If DFSORT is available, the input data set is not partitioned and the record format is not undefined, File Manager automatically uses the DFSORT COPY option to perform simple data set to data set copy actions and print actions. You can enhance this processing by supplying a procedure containing DFSORT statements.

Note:

1. File Manager does not use DFSORT COPY if the input data set is VSAM, unless there are DFSORT commands in the procedure.
2. You cannot use DFSORT for PDS(E)s.

- **REXX programming statements**

  REXX stands for the REstructured eXtended eXecutor language. REXX is a general purpose programming language, similar to PL/I. It includes extensive parsing capabilities for character manipulation, extensive mathematical capabilities, and numerous built-in functions that perform various processing, searching, and comparison functions.

  REXX statements can be processed one at a time by the REXX interpreter or translated into another form for execution (compiled) like a traditional programming language. The initial overhead of compilation usually results in faster execution than interpretation when a large number of records is being processed. In either case, File Manager is able to exploit the power of REXX when processing REXX statements in a procedure.

  FASTREXX

  For a restricted set of REXX statements, File Manager can process the procedure internally, saving the overhead of invoking REXX, and generally resulting in faster processing of the procedure. Within File Manager, this subset of statements is referred to as FASTREXX.

  In order for the procedure to be executed as FASTREXX, every statement in the procedure must be in the FASTREXX subset, otherwise the whole procedure is executed as REXX.

  If the procedure is not eligible for internal processing, it is either run by invoking REXX, or the File Manager function invoking the procedure is terminated, depending on the use of the *REXXPROC and *FASTREXX statements in the procedure.

Choosing between FASTREXX, REXX, and DFSORT for your procedure

FASTREXX is somewhat more CPU-intensive than DFSORT, but much less so than REXX. It has easier syntax than DFSORT, requires only one language to be learnt, and has more function and flexibility than DFSORT, with performance that is close to that of DFSORT. In general, FASTREXX is the most desirable compromise between performance and function. Be aware though that FASTREXX is a subset of REXX and can degrade into REXX, which is the slowest (but most flexible) of the three procedure languages.

REXX is a complete programming language. As such, it is more flexible, but uses significantly more CPU than DFSORT and FASTREXX. There is a "Compiler and Library for REXX on zSeries" product which can be purchased and used by (among others) File Manager. However, while compiled REXX is definitely more CPU-efficient than ordinary REXX, FASTREXX uses significantly less CPU than compiled REXX.

DFSORT is slightly more CPU-efficient than FASTREXX. However, using DFSORT does have some drawbacks:

- You must use an additional command language (DFSORT's).
- File Manager only uses DFSORT when the data is in sequential files, or on VSAM files if there are DFSORT statements in the procedure. File Manager does not use DFSORT when the data is in PDS(E)s or MQ queues.

You should only use DFSORT for jobs where performance is critical.

File Manager's use of DFSORT:

1. To use DFSORT for I/O or as your procedure language, DFSORT must be installed so that File Manager can use it. There is no IBM charge for using the DFSORT technology, even if your site does not have a DFSORT license but instead uses a competitive sort. There is an installation procedure which allows File Manager to use DFSORT, which is always shipped with z/OS, even though ordinary users cannot use it unless the site has a license for DFSORT.
2. File Manager uses DFSORT for I/O processing whenever possible, regardless of the decision about the user procedure language so, during File Manager installation, File Manager should be enabled to use DFSORT.

Should you combine DFSORT with REXX or FASTREXX processing? If you can do this in such a way as to optimize performance (using careful measurement), the answer may be yes. However, it comes at the cost of an increase in complexity in writing, maintaining, and debugging the user procedure. The guideline of keeping things simple strongly favors using FASTREXX.

Coding DFSORT procedure statements

If you want to use DFSORT statements in a procedure, then the first line in the procedure must contain the string `*FASTPROC` coded in column one. You must then code your DFSORT statements in the subsequent lines, prior to any REXX statements. DFSORT statements are coded in columns 1-71.

File Manager supports only the following DFSORT statements:

- INCLUDE/OMIT
- INREC
- OUTREC
- OUTFIL

No other DFSORT statements are supported, except that DFSORT format comment statements can be interspersed with the DFSORT statements. These statements have an asterisk (*) in column 1; they are ignored by File Manager and are not listed in any output.

You can use the DFSORT INCLUDE and OMIT statements to conditionally select records, INREC and OUTREC statements to reformat data, and OUTFIL statements to write to data sets in addition to the primary output data set. This eliminates the need for REXX processing in many basic scenarios.

The DFSORT statements and the output from DFSORT are only displayed if an error is detected. However, if a listing of the DFSORT control statements and output are especially required, the following File Manager control statement can be used in a batch job:

```
$$FILEM SHOWSORT
```

This must be placed as the first File Manager control statement. The output is written to the SYSPRINT data set.

For general information about writing DFSORT statements and for the details of DFSORT statement syntax, see the *DFSORT Application Programming Guide*.

Pay particular attention to the following points:

DFSORT labels
DFSORT syntax allows a statement label to be coded in column one. This means that operation keywords (such as INCLUDE) must not begin before column two.

DFSORT keywords
DFSORT keywords must be coded in upper case.

Statement continuation
Statements ending with a comma or semicolon are concatenated with the next statement at the first non-blank character.

Statements with a non-blank character in column 72 are concatenated with the next statement. All blanks up to and including column 71 are included in the concatenated statement.

Note: These continuation rules are specific to File Manager, and are a little different from the standard DFSORT statement continuation rules. If you are using DFSORT statements coded for use with the DFSORT product, you might need to modify them to use them in a File Manager procedure.

Specifying data positions in variable-length records
Data positions in DFSORT control statements that refer to variable-length records must allow four extra bytes for the record descriptor word (RDW) preceding the data. For example, the following INCLUDE statement for a fixed-length record:

```
INCLUDE COND=(1,2,CH,EQ,'01')
```

becomes:

```
INCLUDE COND=(5,2,CH,EQ,'01')
```

for a variable-length record.

When File Manager uses DFSORT to process records, all VSAM data sets, except for fixed-length relative record data sets (RRDS's), are treated as variable-length. Non-VSAM data sets are treated as fixed or variable-length, according to the data set specifications.

Using OUTFIL statements
When using OUTFIL statements to write multiple output files, direct one of them to the same file as the File Manager output so any data not needed for the OUTFIL data sets is discarded (see "Copying data" on page 425 for an example).

The following examples demonstrate some simple DFSORT statements.

Example 1

To include all records where columns 1-4 contain the name FRED.

```
*FASTPROC
 INCLUDE COND=(1,4,CH,EQ,C'FRED')
```

Example 2

To process only those records where columns 1-80 are not all blank.

```
                                                                    col 72
*FASTPROC
 OMIT COND=(1,80,CH,EQ,C'                                             X
                                    ')
```

Coding REXX procedure statements

File Manager has no special syntactical requirements for REXX statements coded in procedures, except when combining REXX statements with DFSORT statements (see "Combining DFSORT and REXX statements in a procedure" on page 416), or when invoking FASTREXX processing. In addition to the standard REXX statements, File Manager introduces several new REXX variables and functions that you can use to enhance File Manager processing.

For general information about writing REXX, see the *z/OS TSO/E REXX Reference*. For information about the REXX variables, see "How REXX-enhanced processing works" on page 419. For information about the additional REXX functions, see "External REXX functions" on page 1094.

Note: Users familiar with REXX in the z/OS environment might be accustomed to coding a comment containing the word "REXX" at the start of their REXX programs to enable them to run from the SYSPROC DD data set concatenation. This special comment is *not* required in File Manager procedures.

The following examples demonstrate some simple REXX statements.

Note: REXX is case-insensitive when processing REXX keywords and function and variable names. The use of upper and mixed case in these examples is purely stylistic.

REXX example 1

To change all records into upper case:

```
Upper OUTREC
```

REXX example 2

To reformat text so that it is justified to the left and right margins:

```
OUTREC = Justify(OUTREC,Length(OUTREC))
```

REXX example 3

This example invokes the CHANGE function provided by File Manager to change the first occurrence of a pair of slash characters (/) to a pair of question marks (?), but only if the first two characters of the record contain '01'. Otherwise, it changes the first pair of slashes to exclamation marks (!):

```
If Substr(INREC,1,2) == '01' Then
  OUTREC = Change(OUTREC,'//','??')
Else
  OUTREC = Change(OUTREC,'//','!!')
```

Note: This example can be coded more efficiently in FASTREXX. See FASTREXX example 3.

Using internal (FASTREXX) processing

File Manager attempts to use FASTREXX or REXX processing, depending upon the first *REXXPROC or *FASTREXX statement found in the procedure. If no *REXXPROC or *FASTREXX statement is found, then the REXX statements are processed as if they were preceded by a *REXXPROC statement.

Note:

1. FASTREXX processing is supported for the DSEB function as long as it adheres to the following rules.
2. Default length for packed fields: If you omit the length for a function that refers to a packed field processing, the length is re-calculated for each record processed.

***REXXPROC**
The *REXXPROC statement indicates that the subsequent procedure statements are REXX statements (separating them from any preceding DFSORT statements). The statements are processed internally (using FASTREXX) if possible. If FASTREXX processing is not possible, the REXX statements are processed by invoking REXX.

***FASTREXX (ON)**
The *FASTREXX (or *FASTREXX ON) statement indicates that the subsequent procedure statements are REXX statements (separating them from any preceding DFSORT statements), and should be processed internally if possible. If FASTREXX processing is not possible, the File Manager function terminates, indicating FASTREXX processing was not possible.

***FASTREXX OFF**
The *FASTREXX OFF statement indicates that the subsequent procedure statements are REXX statements (separating them from any preceding DFSORT statements), and should be processed by invoking REXX.

***FASTREXX CHECK**
The *FASTREXX CHECK statement indicates that the subsequent procedure statements are REXX statements (separating them from any preceding DFSORT statements), and should be processed by invoking REXX. However, File Manager indicates whether FASTREXX processing was possible for the procedure.

***FASTREXX NORUN**
The *FASTREXX NORUN statement indicates that the subsequent procedure statements are REXX statements (separating them from any preceding DFSORT statements), and should be checked to determine whether FASTREXX processing is possible for the procedure. The Fil Manager function does not run, but File Manager indicates whether FASTREXX processing is possible for the procedure.

For a procedure to be eligible for FASTREXX processing, it must consist only of:

- Null clauses. Null clauses consist only of blanks or comments. They are ignored when REXX statements are processed internally.
- DO-END, DO WHILE-END, DO UNTIL-END, DO FOREVER-END ITERATE LEAVE
- SELECT-WHEN-OTHERWISE-END clauses
- LABEL and SIGNAL label clause support
- IF-THEN-ELSE instruction clauses whose condition expressions are valid for internal processing. These clauses are supported if the condition expression is

valid for internal processing, and the THEN and ELSE clauses are also valid for internal processing. The following functions can be included in an expression that is to be processed internally:

- FLD
- FLD_CO
- FLD_TM
- FLD_TYPE
- I_LENGTH
- O_LENGTH
- PRTCOUNT
- RECSIN
- RECSOUT
- RECCUR
- TFLD
- FLDI
- FLDO
- TESTC
- TESTN
- DOWN
- UP
- FINDNEXT
- FINDPREV

Refer to "About expression processing" on page 213 for more information about the eligibility of condition expressions for internal processing.

- NOP instruction clauses (these are ignored).
- RETURN instruction clauses.
- Command clauses consisting only of an invocation of one of a limited set of functions. These are:
  - CHG_OUT
  - CHG_VAR
  - FLD_OUT
  - VAR_OUT
  - OVLY_OUT
  - OVLY_VAR
  - PRT_IN
  - PRT_OUT
  - PRT_VAR
  - SET_OLEN
  - TALLY
  - WRITE
  - SETC
  - SETN
  - TOP
  - BOT
  - DOWN
  - UP

- RECCUR
- FINDPREV
- FINDNEXT
- UPDATE

The function performs the requested action, and returns a single blank as the command, which is ignored by the File Manager host command environment.

To be eligible for internal (FASTREXX) processing, all of the arguments passed to a command function must be literals or unassigned symbols. Specifically:

- Nested function invocations are not supported.
- Field reference symbols (#ref) are not supported.
- Assignment clauses are not supported.
- Label clauses are not supported.
- The following symbols are not supported, as they involve implicit assignment:
  - INREC
  - OUTREC
  - RC
  - RESULT
  - SIGL

  Note: Because FASTREXX does not support the use of INREC and OUTREC, you must instead exploit the set of functions that act directly on the input and output record.

The following examples illustrate some simple FASTREXX-eligible statements:

FASTREXX example 1

To change the first byte in every record to a blank:

```
OVLY_OUT(' ',1,1)
```

FASTREXX example 2

To pad every record to a length of 100 with blanks (records longer than 100 bytes would remain unchanged):

```
OVLY_OUT(' ',1,100,'C',' ')
FLD_OUT(1,,1)
```

FASTREXX example 3

This example invokes the CHG_OUT function provided by File Manager to change the first occurrence of a pair of slash characters (/) to a pair of question marks (?), but only if the first two characters of the record contain '01'. Otherwise, it changes the first pair of slashes to exclamation marks (!):

```
If FLD(1,2) == '01' Then
  CHG_OUT('//','??')
Else
  CHG_OUT('//','!!')
```

REXX coding hints and tips

It is wise to make a habit of using “strict comparison” for strings, rather than using REXX's looser comparison.

REXX draws a very important distinction between using “=” for an equality test and using “==”. (Similar comments also apply to < and <<, and other operators, but the most important issue is equality testing.)

When you use "==", REXX treats the arguments as strings and compares the arguments unchanged. However, when you use "=", REXX attempts to transform the arguments prior making the comparison. REXX first tries to convert both arguments to numeric values; if this fails, it treats the arguments as strings, stripping leading and trailing blanks and then, if the resulting strings are of unequal length, padding the shorter string on the right with blanks.

For FM REXX programming, a good rule of thumb is to use "==" for all string comparisons, and to use "=" only with care for numeric comparisons.

When you use the strict equality operator ("=="), you must pad literals with blanks as required. For example, if the first 10 bytes of each record contain an uppercase name with trailing blanks, then the result of the following statement is true:

```
FLD(1,10) = 'FRED'
```

but the following statement is false:

```
FLD(1,10) == 'FRED'
```

To avoid the problem shown by the second example, (but still keeping with the preferred use of "==" rather than the looser "="), you could code either of the following alternatives:

```
STRIP(FLD(1,10)) == 'FRED'
FLD(1,10) == 'FRED      ' (six trailing blanks)
```

Ultimately, the last choice is best because it allows FASTREXX processing, which the use of the STRIP function does not.

Here are some examples of possibly unexpected results:

REXX	Result
`IF 'A ' =  ' A'`	True
`IF 'C140'x = '40C1'x`	True
`IF 'A ' ==  ' A'`	False
`IF '01' =  '1 '`	True
`IF '001' =  '1E0'`	True (1E0 is scientific notation)
`IF '+ 1.00'  = '   1E0'`	True
`IF '12345678901' = 12345678902`	True (it's longer than the default REXX numeric precision for integers)

Coding end of file procedure

To code an end of file procedure you need to use the *EOFPROC statement, as shown here:

***EOFPROC**

If this statement is coded, then all the subsequent REXX statements are treated as procedural statements to be run once at normal completion of the function processing. This could be end of file, or when a processing limit has been been reached, or after a STOP condition has been issued from a REXX procedure. The input and output records will be positioned at the last record processed by the function. This end of file procedure is only run for the functions DSC, DSP, DSU, and DSEB. The procedure is ignored for any other functions. If you have previously coded an *FASTREXX ON statement, then this procedure must be eligible for FASTREXX processing, as described in the given example, to avoid syntax

errors. If internal FASTREXX processing is not required, then File Manager chooses either the internal or REXX processing, for the end of file procedure. So one procedure could be running under REXX, while the the other could be FASTREXX. See given examples for usage scenarios.

Example 1.

Perform a copy with a record processing REXX procedure and end of file procedure.

```
$$FILEM DSC PROC=*
SAY 'NORMAL PROCESS RECORD NUMBER' SUBSTR(INREC,5,2)
*EOFPROC
SAY 'END OF FILE PROC INREC' SUBSTR(INREC,8)
```

Example 2.

Perform a copy with a FASTREXX end of file procedure.

```
$$FILEM DSC PROC=*
*EOFPROC
SETC(PRTVAR,' ')                                    /* CLEAR IT */
SETC(INSUB,'&ZINREC(8)')                            /* SUBSTR   */
OVLY_VAR(PRTVAR,'END OF FILE PROC INREC ',1)        /* LITERAL  */
OVLY_VAR(PRTVAR,'&INSUB',0)                         /* EOF VALUE*/
PRT_VAR(PRTVAR)                                     /* PRINT IT */
```

Example 3.

Perform a copy using *FASTPROC, *REXXPROC, and *EOFPROC statements.

```
$$FILEM DSC PROC=*
*FASTPROC
OMIT    COND=(68,2,CH,EQ,C'XX')
*REXXPROC
SAY 'NORMAL PROCESS RECORD NUMBER' SUBSTR(INREC,5,2)
*EOFPROC
SAY 'END OF FILE PROC INREC' SUBSTR(INREC,8)
```

Combining DFSORT and REXX statements in a procedure

When you want your procedure to contain both DFSORT and REXX statements, then you must code the DFSORT statements before the REXX statements, and the REXX statements must be separated from the DFSORT statements by a line containing the string `*REXXPROC` or one of the `*FASTREXX` statements coded in column one. For more information about how enhanced processing interacts with DFSORT and REXX, see “How enhanced processing works” on page 419.

While the DFSORT *FASTPROC statements must be placed before the REXX statements, when a combination of DFSORT and REXX statements are used, the order of execution is:

1. DFSORT OMIT or INCLUDE statements, INREC statements and OUTREC statements, in that order.
2. REXX statements
3. DFSORT OUTFIL statements

Example

This example combines DFSORT and REXX statements in one procedure to change all records, where columns one and two contain '01', into upper case and discard all other records:

```
*FASTPROC
 INCLUDE COND=(1,2,CH,EQ,C'01')
*REXXPROC
 Upper OUTREC
```

Supplying a procedure to enhance File Manager processing

How you supply a procedure to enhance File Manager processing depends on whether you are using a File Manager panel, running File Manager in batch, or invoking File Manager from a REXX program.

Supplying a procedure when using a File Manager panel

If you are using one of the File Manager panels that supports enhanced processing, then you use the **Use proc** (or **Use REXX proc**) field on the panel to supply the procedure:

```
/ Use proc ________
```

If you enter an * (asterisk) as the *proc* name, then File Manager displays an ISPF edit panel where you can enter a temporary procedure for one time use. Use this method only for short, ad hoc procedures. If you want, you can use ISPF's CREATE command to save the procedure for later use.

Alternatively, you can allocate an FMNEXEC DD that identifies a PDS where your saved procedures reside. If you specify a member name, then File Manager either edits a new member or runs with an existing member. Specifying a blank or a pattern other than * displays a member selection list of the PDS allocated to FMNEXEC. Concatenated FMNEXEC data sets are not supported under ISPF.

For example, if you store your procedures in a PDS called 'USERID.FMNEXEC' then, before using the **Use proc** field, you must issue a TSO ALLOC command similar to the following:

```
TSO ALLOC DD(FMNEXEC) DSN('USERID.FMNEXEC') SHR
```

If errors are detected in the procedure when the function is run, File Manager displays the errors in a message box. Correct the errors and rerun the function.

Supplying a procedure when running File Manager in batch

If you are using one of the File Manager functions that supports enhanced processing, then you use the PROC parameter to supply the procedure.

The procedure that you use to enhance File Manager processing can either be a member of the PDS allocated to ddname FMNEXEC, or it can be coded in-line (by specifying an asterisk as the procedure name).

To use a procedure stored in a member of a PDS:

1. Code an `FMNEXEC DD` statement that specifies the name of the PDS that contains the member.
2. Code `PROC=`*`member-name`* in the File Manager control statements.

For example, to use the REXX procedure stored in FMNUSER.EXEC(EXAMPLE):

```
//jobcard
//*
//FILEMNGR PROC
//FILEMAN  EXEC PGM=FILEMGR
//STEPLIB  DD  DISP=SHR,DSN=FMN.V1R1M0.SFMNMOD1
//SYSPRINT DD  SYSOUT=*
//SYSABEND DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//         PEND
//*
//*   Run DSU function with an existing REXX procedure.
//*   The REXX procedure used is in member EXAMPLE in FMNUSER.EXEC.
//*   Member BATCH of PDS FMNUSER.FMN.CNTL is updated.
//*   The updated member is printed.
//*
//EXAMPLE    EXEC FILEMNGR
//FMNEXEC  DD  DISP=SHR,DSN=FMNUSER.EXEC
//INDD     DD  DISP=SHR,DSN=FMNUSER.FMN.CNTL
//SYSIN    DD  *
$$FILEM DSU INPUT=INDD,MEMBER=BATCH,PROC=EXAMPLE
$$FILEM DSP DSNIN=FMNUSER.FMN.CNTL(BATCH)
/*
```

To use an inline procedure:

1. Code PROC=* in the File Manager control statements.
2. Code your DFSORT or REXX statements in the SYSIN data stream, following the PROC=* statement.
3. Terminate the DFSORT or REXX statements with a line containing a slash and a plus sign (/+) in columns 1–2.

For example:

```
//jobcard
//*
//FILEMNGR PROC
//FILEMAN  EXEC PGM=FILEMGR
//STEPLIB  DD  DISP=SHR,DSN=FMN.V1R1M0.SFMNMOD1
//SYSPRINT DD  SYSOUT=*
//SYSABEND DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//         PEND
//*
//*   Run DSU function with a REXX EXEC specified as an in-line proc.
//*   Member BATCH2 of PDS FMNUSER.FMN.CNTL is updated.
//*   The updated member is printed.
//*
//EXAMPLE2    EXEC FILEMNGR
//INDD     DD  DISP=SHR,DSN=FMNUSER.FMN.CNTL
//SYSIN    DD  *
$$FILEM DSU INPUT=INDD,MEMBER=BATCH2,PROC=*
 outrec = change(outrec,'//','??');
/+
$$FILEM DSP DSNIN=FMNUSER.FMN.CNTL(BATCH2)
/*
```

When coding an inline procedure, do not begin comments in column 1; if you do, the /* is interpreted as an end-of-data delimiter.

Supplying a procedure when invoking File Manager from a REXX program

You can specify a PROC parameter to enhance File Manager processing even when you are invoking the File Manager function from a REXX program.

To use a procedure stored in a member of a PDS:

1. Allocate an FMNEXEC DD that identifies a PDS where your saved procedures reside.
2. Code **PROC=***member-name* in the File Manager function invocation.

For example, to use the procedure stored in USERID.FMNEXEC(TOTALS):

```
/* REXX program to print a data set, with enhanced processing */
/* as defined in the "nested" REXX procedure named TOTALS     */
"ALLOC DD(FMNEXEC) DSN('USERID.FMNEXEC') SHR"
:
FILEMGR "$DSP DSNIN='FMNUSER.TEST.KSDS1',PROC=TOTALS"
:
"FREE DD(FMNEXEC)"
```

Note: *The TSO commands ALLOC and FREE can also be issued outside the REXX program.*

To enter a procedure from the terminal

1. Code **PROC=*** in the File Manager function invocation.
2. When you run the REXX program and it invokes the File Manager function, File Manager prompts you to enter procedure statements with the following messages:

   ```
   SYSIN DD is allocated to terminal
   Enter card input (EOD=NULL REPLY):
   ```

   Type your procedure statements, pressing the Enter key at the end of each line.
3. Enter /+ as the last line to complete the procedure and indicate end of data. File Manager then runs the requested function using the entered procedure.

How enhanced processing works

File Manager implements enhanced processing by exploiting the power of DFSORT and REXX to extend the basic operation of File Manager functions and panels. Various details of how File Manager interacts with these external products is discussed in the following sections.

How REXX-enhanced processing works

File Manager defines two special REXX variables, INREC and OUTREC, that you can use in the REXX statements that you supply to perform enhanced processing. When the File Manager function or panel invokes REXX, the contents of each input record selected for processing are passed in both INREC and OUTREC. When REXX is invoked, the contents of INREC and OUTREC are identical, unless you are using the DSC function or the Copy Utility (option 3.3) with an output template that performs field mapping to reformat records. If you are using DSC or the Copy Utility to reformat records then, when REXX is invoked, INREC contains the input record, and OUTREC contains the reformatted output record.

The INREC variable is intended to be used as a reference variable. Any changes made to it are ignored by File Manager. The OUTREC variable can be updated by the procedure, and (unless you “drop” the record from further processing, as described below) when REXX processing completes, is passed back for processing by the File Manager panel or function that you are enhancing. For example, the following code processes a data set containing records with a type indicator in the first two bytes. Records with type '01' are all 80-bytes long and are to be passed on

for output without change. Records with type '02' are variable in length, all less than 80 bytes long, and contain data divided into two sections (each at most 40 bytes long) by a slash character ('/'). These records are reformatted into two 40-byte halves that are concatenated to make 80-byte records to be passed on for output.

```
/* Reformat varying length records as fixed
   using an arbitrarily located delimiter   */
If Substr(inrec,1,2) == '02' Then Do
  Parse Var inrec left '/' right
  outrec = Left(left,40) || Left(right,40)
End
```

If the value of the OUTREC variable is longer than the record length allowed in the data set specified by the function or panel you are using, then it is truncated. If the value of OUTREC is shorter than the record length, it is padded using the character specified by the **PAD** field on the Set Processing Options (option 0) panel (if you are enhancing a panel) or the character specified by the PAD parameter of the SET function (if you are enhancing a function).

How DFSORT-enhanced processing works

File Manager extracts the DFSORT control statements, if any, from the procedure, and invokes DFSORT to perform the required input and output operations. The control statements are passed to DFSORT and processed by that product according to its rules. If template processing is not required, and the procedure does not contain any REXX statements, then the entire operation is performed under the control of DFSORT.

If File Manager needs to get control during the operation, to perform template processing or execute REXX statements contained in the procedure, then it does this via DFSORT's E35 exit. This means that all of the processing for the INCLUDE, OMIT, INREC, and OUTREC statements is performed before File Manager regains control for each record. When File Manager regains control, it first applies the record identification and selection criteria associated with the input template, and the mapping (for the REXX OUTREC variable) associated with the output template, then invokes REXX to process REXX procedure statements, if any, and finally it applies any formatting operations associated with the output template.

When File Manager returns control to DFSORT any OUTFIL control statements found in the procedure are processed. This means that your OUTFIL statements must be coded to take into account not only the effect of any INREC or OUTREC statements coded in the procedure, but also the effect of any template or REXX processing that might also have been performed. Correspondingly, any templates used or REXX statements processed must allow for the effect of any INREC or OUTREC statements coded in the procedure.

Enhanced processing modes

There are two primary modes of enhanced processing:

- Enhanced record processing
- Enhanced member processing

A single execution of a File Manager utility can perform one or the other mode of processing, but not both.

In both enhanced record and enhanced member processing, individual records are presented to the user procedure. In enhanced record processing, the user procedure can make decisions about what to do with each record. In enhanced member

processing, the user procedure cannot change individual records, but it can make decisions about how to process the entire member.

Enhanced record processing

The following sections describe how File Manager performs enhanced record processing for each utility function.

Copy Utility (option 3.3) or Data Set Copy function (DSC)

In DSC record processing, each record is read from the input file. The input record is passed to the user procedure in an "input record buffer", and also in an initially identical "output record buffer". The user procedure can take a number of actions or, as long as they are mutually compatible, any combination of actions, based on data inside the record and from other sources. The user procedure then returns control to File Manager. At this point, File Manager copies the output record from the user procedure to the primary output file (assuming that those are the instructions from the user procedure).

For each input record, the procedure can:

- Leave the output record unchanged and return it for copying to the output file. (RETURN command)
- Modify some or all of the output record and then return if for copying to the output file. (RETURN command)
- Choose to not copy the output record to the output file. (RETURN DROP command)
- Stop processing the input file and therefore stop writing to the output file. (RETURN STOP and RETURN STOP IMMEDIATE commands)
- Format new records based on any data in the input record and write one or more new records to one or more output files other than the primary output file. (WRITE command)

A user procedure is optional for DSC.

Print Utility (option 3.2) or Data Set Print function (DSP)

The record processing for DSP is very similar to that for DSC. However, whereas DSC writes the user procedure's output record to an output file, DSP formats the output record as requested for printing and sends it to the print destination.

A user procedure is optional for DSP.

Data Set Update (DSU) function

DSU record processing differs from DSC and DSP in that the records in the data set are not copied. Instead, records can be optionally updated by the user procedure. The user procedure is invoked after File Manager reads a record. The user procedure's input and output record buffers are set to the record as present in the file.

For each input record, the procedure can:

- Leave the output record unchanged. If the output record is unchanged, File Manager does not update the file record. (RETURN command)
- Modify some or all of the output record and then return it for updating in the file. (RETURN command)
- Explicitly choose to not update the file record. (RETURN DROP command)
- Stop processing the input file. (RETURN STOP and RETURN STOP IMMEDIATE commands)

- Format new records based on any data in the input record, and write one or more new records to one or more output files other than the primary output file. (WRITE command)

There are a few restrictions on DSU. The main restriction is that DSU performs an "update in place". That is, the updated data set record must be the same length as the record prior to updating.

A user procedure is required for DSU.

Data Set Edit in Batch (DSEB) function
DSEB record processing is similar to that for DSU. Like DSU, the DSEB user procedure can perform an "update in place" at any record. However, in DSEB, the user procedure can only traverse the records in the data set by including navigational commands such as UP() and DOWN(). As a result, the input data set records can be traversed and updated in any order.

A user procedure is required for DSEB.

Find/Change Utility (option 3.6) or Find/Change function (FCH)
Normal FCH usage is without a user procedure. In this case, FCH primary commands are used to search for strings (and possibly change them) in a data set or library.

An FCH user procedure can perform complex logic instead of a simple string match on a record. It can then use RETURN or RETURN DROP to indicate whether the record should be reported as selected ("found") in the FCH output report. The user procedure can also change the record by RETURNing a changed output record; in this case the record is always also considered found or selected.

Enhanced member processing

You can enhance the following utilities with member processing:
- Copy Utility (option 3.3) or Data Set Copy function (DSC)
- Print Utility (option 3.2) or Data Set Print function (DSP)

The key concept is that the user procedure is presented with each record in the member. The user procedure tests each record for some conditions and then tells File Manager whether or not the entire member should be copied or printed.

Member processing always requires a user procedure. The commands, RETURN PROCESS MEMBER and RETURN DROP MEMBER, enable the user procedure to indicate respectively whether or not the current utility operation (print or copy) should be performed on the member. If the user procedure makes no decision, File Manager continues to pass records to the user procedure by using the RETURN command.

For member processing, you must specify the MEMPROC option, as well as a default behavior (PROCESS or DROP). If the entire member is read and no decision is made by the user procedure (that is, there is no RETURN PROCESS MEMBER or RETURN DROP MEMBER command returned), then the default behavior (PROCESS or DROP) is used.

Changes made by the user procedure to the output record are ignored during member processing.

Using templates with enhanced processing

If you specify a template for DSEB and DSU, it is ignored, except for calls in a REXX program to the REXX external function PRINT that specify TABL or SNGL format. For details, see "PRINT" on page 1135.

For the other functions and panels that you can enhance, any record selection by the template is performed before the procedure is run; only those records that the template selects are passed to the procedure for further processing.

Regardless of whether or not the template you are using selects only certain fields, the INREC and OUTREC variables defined for the REXX environment contain complete records with all fields. However, after the REXX statements in the procedure have been processed (or when you call the PRINT function in the procedure with TABL or SNGL format), the record is printed using only the fields selected by the template.

You cannot specify a template for use with FCH or Find/Change Utility (option 3.6).

For information on using templates, see Chapter 4, "Creating and editing templates," on page 127.

Working with an entire data set (DSEB only) or per record

Except for the DSEB function, the REXX statements that you supply are run against each input record in sequence, either until the end of the input data set is reached, or until a RETURN STOP (or RETURN STOP IMMEDIATE) statement is executed. However, the REXX statements you supply with a DSEB function are run only once, for the entire input data set. In the REXX statements for a DSEB function, you can use various external REXX functions (described later in this chapter) to move between records in the input data set.

Dropping a record from further processing

Except for the DSEB function, if the REXX code returns normally after processing a record, then that record is considered to be "selected", and the contents of OUTREC are used as the output record for copying, printing, updating, or inclusion in a Find/Change report (according to the function or panel you are enhancing). However, if a RETURN DROP (or RETURN STOP IMMEDIATE) statement is executed, then the current record is "dropped" from further processing by File Manager, and is not copied, printed, updated, or included in a Find/Change report.

For Data Set Copy (DSC) and Data Set Print (DSP), the DFSORT OMIT statement can be used to "drop" a record form further processing.

For DSEB, the contents of OUTREC are only written to the output data set when you call the UPDATE function. If you leave the procedure or move to another record before calling the UPDATE function, then any changes to the current OUTREC are lost.

Performance tips

Make your REXX procedures FASTREXX eligible
Look for a way to code your requirements using FASTREXX wherever

possible. You can test your existing procedures to see if they are FASTREXX eligible by using the *FASTREXX CHECK or *FASTREXX NORUN statements.

Use DFSORT for performance-critical jobs
DFSORT processing is more efficient than FASTREXX and significantly more efficient than REXX. See "Choosing between FASTREXX, REXX, and DFSORT for your procedure" on page 408.

For maximum efficiency when mixing DFSORT and REXX statements, where appropriate:

- Use DFSORT in preference to REXX when you can achieve the same result from either.
- Use DFSORT INCLUDE or OMIT statements to restrict the amount of REXX processing to be done by restricting the number of calls to REXX.
- Write multiple output files by using DFSORT OUTFIL rather than REXX write statements.
- Be prepared to use two passes through the data, rather than one, if that would make processing more efficient.
- Make the REXX part of your procedures FASTREXX eligible wherever possible.

Do not use EXITs in REXX
Use RETURN to terminate all your REXX processing. Do not use EXIT statements. EXIT statements in REXX used with File Manager cause excessive CPU usage.

Run with the REXX compiler
Compiled REXX runs considerably faster than interpreted REXX. However, because File Manager modifies the REXX you supply to run in the File Manager environment, you cannot supply pre-compiled REXX code. Instead, by supplying the DD card for the REXX compiler library, you instruct File Manager to compile your REXX before running it. This approach reduces CPU usage and run time.

File Manager runs the compiler when you allocate the data set containing your REXX compiler to the DD name FMNRXCMP. For example, in batch, use the following JCL DD statement:

```
//FMNRXCMP DD DSN=rexx.compiler.loadlib,DISP=SHR
```

When running a procedure from a panel, you can enter a TSO ALLOC command on the Command line. For example,

```
TSO ALLOC DD(FMNRXCMP) DSN('rexx.compiler.loadlib') SHR
```

The REXX compiler has no effect on DFSORT or FASTREXX processing.

REXX and DFSORT Examples

The examples in this section demonstrate the use of REXX and DFSORT statements to achieve common tasks.

Printing data

The following excerpt from a batch job uses the DSP print function to:

- Skip the first 900,000 records
- Exclude all records from further processing where the first 4 characters are not "CHAX"

- Print the next 1000, non-excluded records, formatted using template TEMPL98

```
⋮
//SYSIN    DD  DATA
$$FILEM DSP FORMAT=TABL,
$$FILEM POSITION=900000,
$$FILEM NLRECS=1000,
$$FILEM TCIN=FMNUSER.FMNAFDATA.TEMPLATE(TEMPL98),
$$FILEM DSNIN=FMNUSER.FMNAFDAT.SAMP,
$$FILEM PROC=*
*FASTPROC
 OMIT COND=(1,4,CH,NE,C'CHAX')
+/
⋮
```

Copying data

By default, the DSC (Data Set Copy) function simply copies the contents of one data set to another (using DFSORT if available). The following excerpt from a batch job enhances the DSC function to:

- Include in the input stream only those records whose first two characters are “01” or “02”.
- Add two lines to SYSPRINT that tally the total values of the “salary” (a 4-byte packed decimal field found at position 27) and “month 1 payment” (a 4-byte binary field found at position 31) fields in the “01”-type records.
- In each output record, change the first occurrence of “Grant Smith” to “Fred Bloggs”.
- Write only those records whose first two characters are “02” to the default output data set (DDOUT), adding a sequence field in 1-6 and shifting the rest of the data over.
- Write to another data set (OUT01) only those records whose first two characters are “01”, unchanged.
- Print the first ten output records (to SYSPRINT) in hexadecimal format.

```
⋮
//DDIN     DD DSN=FMNUSER.FMNAFDAT.SAMPMVS,DISP=SHR
//DDOUT    DD DSN=FMNUSER.FMNAFDAT.SAMP02,DISP=SHR
//OUT01    DD DSN=FMNUSER.FMNAFDAT.SAMP01,DISP=SHR
//SYSIN    DD *
$$FILEM DSC INPUT=DDIN,
$$FILEM     PROC=*
*FASTPROC
 INCLUDE COND=(1,2,CH,EQ,C'02',OR,1,2,CH,EQ,C'01')
 OUTFIL FNAMES=DDOUT,INCLUDE=(1,2,CH,EQ,C'02')
 OUTREC=(SEQNUM,6,ZD,1,74)
 OUTFIL FNAMES=OUT01,INCLUDE=(1,2,CH,EQ,C'01')
*REXXPROC
 outrec = change(outrec,'Grant Smith','Fred Bloggs')
 /* Print the first 10 output records in hex */
 if prtcount() < 10 then print(outrec,'hex')
 if fld(1,2) == '01' then do
   tally(27,4,'P','Salary Total')
   tally(31,4,'B','Total Month 1 Payment')
 end
/+
/*
⋮
```

Updating records in a data set

The following example searches every member of the partitioned data set FMNUSER.TESTPDS for records that contain both "JAMES" (starting at column 6) and "BROWNE" (starting at column 20), then changes "JAMES" in those records to "JIMMY".

```
⋮
$$FILEM DSU DSNIN='FMNUSER.TESTPDS',
$$FILEM  MEMBER=*,
$$FILEM  PROC=*
if fld(1,5) == 'JAMES' & fld(20,6) == 'BROWNE' then
   CHG_OUT('JAMES', 'JIMMY', 0)
/+
⋮
```

Editing a data set

The following example shows how you can use the DSEB (Data Set Edit Batch) function to edit a data set from a batch job (this example performs the editing "on itself"):

```
⋮
$$FILEM DSEB DSNIN=FMNUSER.JCL(FMNDSU4),
$$FILEM  PROC=*

bot()                     /* Go to the last record */
rc = findprev('/+')       /* Find the input sentinel */
If rc ¬= 0 then do        /* We found it ok */
  say 'We found it at' rc
  up(1)                   /* backup a record */
  outrec = change(inrec,'fred','john')    /* change fred to john */
  say 'inp-'inrec
  say 'out-'outrec
  update()                          /* and do the update */
  end

rc = findprev('sentinel',40)
If rc ¬= 0 then do
  outrec = change(inrec,'sentinel','marker  ')
  update()
  end

 /* this is a comment with fred in it */
/+
⋮
```

Examples of tasks that can be performed easily using FASTREXX

Some of the tasks you can perform easily with FASTREXX include:

- Record editing.
- Writing one or more optionally edited records to one or more output files, depending on data in the input record.
- Determining whether to copy or print a member based on some condition inside the member.

The following sections provide examples of each of the above tasks and how FASTREXX can be used to perform them.

Note: For simplicity, the examples shown assume Data Set Copy is being performed.

Record Editing

Record editing can include any combination of:
- Rearranging fields of certain records.
- Adding constants or text to certain records.
- Removing fields from certain records.
- Dropping whole records from the output based on the contents of the record.

Example:

Rearrange a record of type "A" (as designated in the first byte) to be a record of type "Z", by reversing the first field of 20 bytes (first name) and the next field of 30 bytes (last name). Then drop the next 4 bytes (52 thru 55) and include the rest of the record. Also, drop all records of type "B".

FASTREXX:

```
If FLD(1,1) = "B" Then RETURN DROP
If FLD(1,1) = "A" Then Do
   SET_OLEN( 0 )
   OVLY_OUT( "Z", 1, 1 )
   FLD_OUT( 22, 30 )
   FLD_OUT( 2, 20 )
   FLD_OUT( 56 )
   End
```

REXX:

```
InitChar = SUBSTR( INREC, 1, 1 )
If InitChar = "B" Then RETURN DROP
If InitChar = "A" Then Do
   OUTREC = OVERLAY("Z" || FLD( 22, 30 ) || FLD( 2, 20 ), OUTREC, 1 )
   End
```

DFSORT:

Note:

1. Requires DFSORT PTFs UQ95214 and UQ95213
2. This only works on sequential input files.

```
*FASTPROC
 OMIT COND=(1,1,CH,EQ,'B')
 OUTFIL IFTHEN=(WHEN=(1,1,CH,EQ,'A'),
      OVERLAY=(1:'Z',22,30,2,20,52)),
   IFTHEN=(WHEN=NONE,
      BUILD=(1)))
```

Writing one or more optionally edited records to one or more output files, depending on data in the input record

This can include:

- Splitting an input file into multiple output files based on the contents of the record

Example:

Split the input file into multiple output files based on the packed value in bytes 1-4. If the value is greater than 100, put the record in DD OV100; if it is less than 10, put the record in LT10; and if the value is in the middle, put the record in MIDDLE. If the value in bytes 1-4 is not packed, put the record into DD ERROR.

Note: In the FM (REXX and FASTREXX) solutions, there is potential for you to make an error. File Manager does not open any of the secondary output DDs for output until there is a WRITE command to that DD executed.

For example, suppose there is never a WRITE to the file ERROR and that ERROR is allocated NEW in the DD card, with BLKSIZE=0 for DFSMS to determine. After DSC executes with the control cards shown below, ERROR will not have been opened, so the BLKSIZE will not have been set acceptably by DFSMS and so it will still be zero. All attempts to read ERROR will generate errors. Therefore, prior to this File Manager utility operation, you should create empty secondary output data sets. You can use File Manager Data Set Generate (DSG) with NLRECS=0.

File Manager always opens the primary DSC output data set. The discussion here only applies to secondary (WRITE) output data sets.

FASTREXX:

```
$$FILEM DSG OUTPUT=ERROR,NLRECS=0
$$FILEM DSG OUTPUT=OV100,NLRECS=0
$$FILEM DSG OUTPUT=LT10,NLRECS=0
$$FILEM DSG OUTPUT=MIDDLE,NLRECS=0
$$FILEM DSC PROC=*
 If \ FLD_TYPE(1,4,"P") Then WRITE( ERROR )
 Else If FLD(1,4,"P") > 100 Then WRITE( OV100 )
 Else If FLD(1,4,"P") < 10 Then WRITE( LT10 )
 Else WRITE( MIDDLE )
 RETURN DROP
```

REXX:

(Same as FASTREXX)

Determining whether to copy or print a member based on some condition inside the member

Example:

Copy all members which contain the character string “Hello Mom” (with any letter in upper or lowercase).

FASTREXX:

```
/* Requires option MEMPROC=DROP */
If FLD_CO( 1, 0, "U", "HELLO MOM" ) Then RETURN PROCESS MEMBER
```

REXX:

(Same as FASTREXX)

DFSORT:

Not applicable.

Relative positioning support

The following sections show how you can use enhanced processing with the relative positioning support provided by File Manager.

Input record relative positioning

File Manager allows a FASTREXX program to search an input record and use the resulting successful search position to make other tests at a fixed offset from the located string. You can also perform this task in full REXX, using REXX variables and non-FASTREXX functions. However, it executes more efficiently if done in FASTREXX using relative positioning.

File Manager maintains an internal “current position” in the input buffer, and also sets the value in the REXX variable, INPOS. (Note: you cannot use REXX variables in FASTREXX.)

To set the current position in the input buffer, use the function FLD_CO (Field Contains). If the search operation is successful, the current position is set to the starting byte of the located string.

You can refer to the input buffer position wherever an input or output buffer position is required in a FASTREXX function. You can adjust it positively (for example, P7 is the current position plus 7 bytes), or negatively (for example, N2 is the current position minus 2 bytes).

Example

Print all members of a JCL library which execute a program with a name like %%DBG* (any 2 initial characters and any following allowed). The way this is implemented is to test if the JCL record contains " EXEC PGM=" and then test the program name for "DBG" at a relative position.

FASTREXX:

```
If FLD_CO(1, 0, C, " EXEC ") ,
 & FLD_CO(P4, 0, C, " PGM=") ,
 & FLD(P7,3) = "DBG" Then
   RETURN PROCESS MEMBER
```

REXX:

```
P = Pos( " EXEC ", INREC, 1 )
If P = 0 Then RETURN
P = Pos( " PGM=", INREC, P )
If P = 0 Then RETURN
If Substr( INREC, P+7, 3 ) = "DBG" Then
   RETURN PROCESS MEMBER
```

DFSORT:

(Can't be done.)

Output record relative positioning

File Manager maintains an internal “current position” in the output record, and also sets the value in the REXX variable, OUTPOS. After any operation that modifies the output record, File Manager resets the value in OUTPOS to point to the next byte position after the most recent change to the output record. By using the value held in OUTPOS, you can easily append fields to the output record.

Example

Rearrange the input record fields in the output record. The first step is to “clear” the output record by setting its length to zero. Then fields and constants are

appended to the output record. The only field and length references needed are in the input record. When no reference is made to an output position, it defaults to the current output position.

FASTREXX:

```
SET_OLEN( 0 )
OVLY_OUT('First name: ')
FLD_OUT(31,20)
OVLY_OUT('Last name: ',P5)  /* Skips 5 bytes, filled by PAD character */
FLD_OUT(1,30)
```

Part 2. File Manager Reference

Chapter 14. Panels and fields

This section of the manual lists all of the File Manager panels, provides a definition for each field in the panel and, where applicable, lists the value ranges that are valid for each field.

Summary of File Manager panels

The table below lists each File Manager panel. For information on using the panels, see the topic in the Panel column. For information about what you use the panel for, see the topic in the Usage column.

For most File Manager panels, there is an equivalent File Manager function that you can use when programming batch jobs, REXX procedures or TSO clists. These are listed in the "Equivalent function" column. For information on programming with File Manager functions, see Chapter 12, "Introduction to programming with File Manager functions," on page 397.

Table 11. File Manager panels

Option	Description	Panel	Usage	Equivalent function
0 0.1–0.9	**Set Processing Options** displays and changes parameters that affect File Manager behavior.	"Set Processing Options panel" on page 661	"Setting your default processing options" on page 41	SET "SET (Set Processing Options)" on page 1047
1	**View** allows you to display and edit data but without the ability to save any changes. You can scroll through the data, search for a line number or string, hide records, and copy data to the clipboard.	"View panel" on page 707	Chapter 3, "Viewing and changing data sets," on page 49	DSV[1] "DSV (Data Set View)" on page 998
2	**Edit** allows you to display and edit data.	"Editor panel" on page 534	Chapter 3, "Viewing and changing data sets," on page 49	DSE[1] "DSE (Data Set Edit)" on page 900
3	**Utility Functions**	"Utility Functions menu panel" on page 703		
3.0	**Set DBCS Format** allows you to specify which columns in a data record are in EBCDIC (the default), DBCS, or mixed format.	"Set DBCS Format panel" on page 651	"Printing DBCS data" on page 305	FMT "FMT (Set DBCS Format)" on page 1024
3.1	**Data Create Utility** allows you to create sequential data sets, VSAM data sets or PDS members, and initialize their field values.	"Data Create Utility panel" on page 515	"Creating data sets and records" on page 242	DSG "DSG (Data Set Generate)" on page 922
3.2	**Print Utility** allows you to print data.	"Print Utility panel" on page 620	"Printing from File Manager" on page 299	DSP "DSP (Data Set Print)" on page 968

Table 11. File Manager panels (continued)

Option	Description	Panel	Usage	Equivalent function
3.3	**Copy Utility** allows you to copy data from any partitioned, sequential or VSAM data set to any other partitioned, sequential or VSAM data set, with optional record selection and field-level reformatting.	"Copy From panel" on page 491	"Copying data sets" on page 253	DSC "DSC (Data Set Copy)" on page 875
3.4	**Catalog Services** displays or prints catalog information and lets you work with catalog entries.	"Catalog Services panel" on page 464	"Managing catalog entries" on page 311	SCS "SCS (Catalog Services)" on page 1045
3.5	Work with **VTOC** displays or prints a list of the data sets on a disk volume obtained from the disk VTOC.	"Display VTOC panel" on page 523	"Displaying a Volume Table of Contents (VTOC)" on page 325	DVT "DVT (Display VTOC)" on page 1006
3.6	**Find/Change Utility** allows you to search for or change a string in a partitioned, sequential or VSAM data set.	"Find/Change Utility panel" on page 574	"Finding and changing data in multiple PDS members" on page 270	FCH "FCH (Find/Change)" on page 1015
3.7	**AFP Print Browse** provides a formatted display of an Advanced Function Printing (LIST3820) document.	"AFP Print Browse panel" on page 442	"Browsing AFP Print documents" on page 329	APB[1]
3.8	**Memory Browse** displays user storage in dump format.		"Browsing your user storage" on page 329	MB[1]
3.9	**Print Browse** browses print output.		"Printing from File Manager" on page 299	PB[1]
3.10	**Load Module Utilities**			
3.10.1	**View Load Module** allows you to display or print a list of the symbols (CSECTs, common sections, entry points, and ZAPs) in a load module (or program object).	"Load Module Information - Selection panel" on page 593	"Viewing load module information" on page 330	VLM "VLM (View Load Module)" on page 1085
3.10.2	**Compare Load Modules** allows you to perform load module comparison,with selection from various compare criteria on module and CSECT level.	"Load Module Information panel" on page 592	"Comparing load modules" on page 297	CLM "CLM (Compare Load Module)" on page 856
3.11	**Compare data** allows you to perform data set comparisons, with optional record selection and field level comparison mapping.	"Compare Utility: "Old" and "New" panels" on page 471	"Comparing data sets" on page 281	DSM "DSM (Data Set Compare)" on page 934
3.12	**Print audit trail report** prints the contents of the audit trail data set.	"Print Audit Trail panel" on page 618	"Printing your Audit Trail Report" on page 308	—
3.13	**Copybook View and Print** displays and prints a copybook or template definition.	"Copybook View and Print: View panel" on page 509	"Copybook View and Print Utility (option 3.13 or 7.2)" on page 193	PBK
3.14 or 9.1	**List Websphere MQ managers and queues** lists Websphere MQ managers and queues.	"WebSphere MQ Managers panel" on page 729	"Working with WebSphere MQ" on page 331	—

Table 11. File Manager panels (continued)

Option	Description	Panel	Usage	Equivalent function
4	**Tape Specific Functions**			
4.1	**Tape Browse** allows you to browse physical records on a tape.	"Tape Browse panel" on page 670	"Tape Browse (option 4.1)" on page 348	TB[1]
4.2	**Tape Data Copy Function**	"Tape Data Copy Functions panel" on page 671	"Tape Data Copy Function (option 4.2)" on page 349	
4.2.1	**Tape to Tape** copies tape files from one tape to another.	"Tape to Tape panel" on page 681	"Tape to Tape (option 4.2.1)" on page 350	TT "TT (Tape to Tape)" on page 1077
4.2.2	**Tape to Tape Reblocked** copies a file from one tape to another, changing the block size and record format.	"Tape to Tape Reblocked panel" on page 683	"Tape to Tape Reblocked (option 4.2.2)" on page 350	TTR "TTR (Tape to Tape Reblocked)" on page 1080
4.2.3	**Tape to Labeled Tape** copies standard labeled tapes.	"Tape to Labeled Tape panel" on page 680	"Tape to Labeled Tape (option 4.2.3)" on page 350	TLT "TLT (Tape to Labeled Tape)" on page 1060
4.2.4	**Tape to VSAM** copies tape records to a VSAM data set.	"Tape to VSAM panel" on page 685	"Tape to VSAM (option 4.2.4)" on page 351	TV "TV (Tape to VSAM)" on page 1081
4.2.5	**Tape to Sequential Data** copies tape records to a sequential data set.	"Tape to QSAM panel" on page 684	"Tape to Sequential Data (option 4.2.5)" on page 351	TS "TS (Tape to Sequential Data)" on page 1074
4.2.6	**VSAM to Tape** copies VSAM records to a tape file.	"VSAM to Tape panel" on page 728	"VSAM to Tape (option 4.2.6)" on page 352	VT "VT (VSAM to Tape)" on page 1089
4.2.7	**Sequential Data to Tape** copies sequential records to a tape file.	"Sequential Data to Tape panel" on page 648	"Sequential Data to Tape (option 4.2.7)" on page 352	ST "ST (Sequential Data to Tape)" on page 1056
4.2.8	**Exported Stacked Volume Copy** copies a logical volume from an Exported Stacked Volume to a physical volume.	"Exported Stacked Volume Copy panel" on page 552	"Exported Stacked Volume Copy (option 4.2.8)" on page 352	EVC "EVC (Exported Stacked Volume Copy)" on page 1011
4.3	**Tape Update** allows you to update tape records on-screen, while copying the data from one tape to another.	"Tape Update panel" on page 686	"Tape Update (option 4.3)" on page 354	TU[1]
4.4	**Tape Record Load** copies a specified number of tape records to another tape, and allows you to alter selected blocks.	"Tape Record Load (option 4.4)" on page 356	"Tape Record Load panel" on page 676	—
4.5	**Tape Print** print tape records, optionally deblocked.	"Tape Print panel" on page 675	"Tape Print (option 4.5)" on page 357	TP "TP (Tape Print)" on page 1063
4.6	**Tape Map** summarizes tape contents for a specified number of files.	"Tape Map panel" on page 673	"Tape Map (option 4.6)" on page 359	TMP "TMP (Tape Map)" on page 1061
4.7	**Create Tape Data** writes test data to tape.	"Create Tape Data panel" on page 513	"Create Tape Data (option 4.7)" on page 360	BT "BT (Create Tape File)" on page 838

Table 11. File Manager panels (continued)

Option	Description	Panel	Usage	Equivalent function
4.8	**Tape Label Display** prints tape labels and a tape label summary.	"Tape Label Display panel" on page 672	"Tape Label Display (option 4.8)" on page 361	TLB "TLB (Tape Label Display)" on page 1059
4.9	**Tape to Tape Compare** compares two tapes byte by byte.	"Tape to Tape Compare panel" on page 682	"Tape to Tape Compare (option 4.9)" on page 362	TTC "TTC (Tape to Tape Compare)" on page 1079
4.10	**Tape Record Scan** scans for a specified string of data in a tape file.	"Tape Record Scan panel" on page 677	"Tape Record Scan (option 4.10)" on page 363	TRS "TRS (Tape Record Scan)" on page 1072
4.11	**Write Tape Mark** writes one or more tape marks at the current position.	"Write Tape Mark panel" on page 737	"Write Tape Mark (option 4.11)" on page 364	WTM "WTM (Write Tape Mark)" on page 1092
4.12	**Initialize Tape**	Chapter 1, "File Manager overview," on page 3	"Initialize Tape (option 4.12)" on page 365	INT "INT (Initialize Tape)" on page 1027
4.13	**Erase Tape** erases a tape from the current position until end-of-tape (EOT).	"Erase Tape panel" on page 551	"Erase Tape (option 4.13)" on page 366	ERT "ERT (Erase Tape)" on page 1010
4.14	**Exported Stacked Volume List** prints the table of contents (TOC) from an Exported Stacked Volume.	"Exported Stacked Volume List panel" on page 553	"Exported Stacked Volume List (option 4.14)" on page 367	EVL "EVL (Exported Stacked Volume List)" on page 1013
4.15	**Tape Positioning Functions**	"Tape Positioning Functions panel" on page 674	"Tape Positioning Functions (option 4.15)" on page 368	
4.15.1	**Backward Space File** moves a tape backward one or more tape files.		"BSF" on page 345	BSF "BSF (Backward Space File)" on page 837
4.15.2	**Forward Space File** moves a tape forward one or more tape files.		"FSF" on page 346	FSF "FSF (Forward Space File)" on page 1025
4.15.3	**Backward Space Record** moves the tape backward one or more records.		"BSR" on page 345	BSR "BSR (Backward Space Record)" on page 837
4.15.4	**Forward Space Record** moves a tape forward one or more tape records or tape marks.		"FSR" on page 347	FSR "FSR (Forward Space Record)" on page 1026
4.15.5	**Tape Rewind** rewinds a tape to the load point.		"REW" on page 347	REW "REW (Tape Rewind)" on page 1044
4.15.6	**Tape Rewind-Unload** rewinds a tape and unloads it.		"RUN" on page 348	RUN "RUN (Tape Rewind-Unload)" on page 1044
5	**Disk/VSAM Data Functions**		Chapter 9, "Disk/VSAM Data Functions," on page 371	

Table 11. File Manager panels (continued)

Option	Description	Panel	Usage	Equivalent function
5.1	**Disk Browse** displays the data structure and contents of records stored on the tracks of the specified physical disk.		"Disk Browse (option 5.1)" on page 371	DB[1]
5.2	**Disk Track Edit** allows you to display and edit records on the specified physical disk.		"Disk Track Edit (option 5.2)" on page 372	DTE[1]
5.3	**Disk Print** prints records from the specified tracks on a physical disk.		"Disk Print (option 5.3)" on page 378	DP "DP (Disk Print)" on page 864
5.4	**Disk Record Scan** searches the specified tracks of a physical disk for a particular string or an EOF record.		"Disk Record Scan (option 5.4)" on page 379	DRS "DRS (Disk Record Scan)" on page 866
5.5	**Write EOF Record** writes a logical EOF record at the specified absolute disk address.		"Write EOF Record (option 5.5)" on page 379	EOF
5.6	**Data Set Extents** displays the beginning and end addresses of the disk extents of a disk data set.		"Data Set Extents (option 5.6)" on page 380	DSX "DSX (Data Set Extents)" on page 1005
5.7	**VSAM Update** allows you to select and then edit a single record in a VSAM data set.	"VSAM Update (option 5.7)" on page 380	"VSAM Update (option 5.7)" on page 380	VU[1]
6	**OAM Functions**		Chapter 10, "OAM Functions," on page 383	
6.1	**Object Directory List** displays or prints a list of Object Access Method (OAM) objects from a collection; if you display the list, you can browse, print, update, or erase any of the listed objects.		"Object Directory List (option 6.1)" on page 383	ODL "ODL (Object Directory List)" on page 1028
6.2	**Object Browse** displays an OAM object.		"Object Browse (option 6.2)" on page 385	OB[1]
6.3	**Object Print** prints an OAM object.		"Object Print (option 6.3)" on page 385	OP "OP (Object Print)" on page 1032
6.4	**Object Update** allows you to display and edit an OAM object.		"Object Update (option 6.4)" on page 386	OU[1]
6.5	**Object Erase** erases an OAM object.		"Object Erase (option 6.5)" on page 386	OE[1]
6.6	**Object Copy Functions**			
6.6.1	**Object to VSAM** copies one or more OAM objects to a VSAM data set.		"Object to VSAM (option 6.6.1)" on page 386	OV "OV (Object to VSAM)" on page 1036
6.6.2	**Object to Sequential Data** copies one or more OAM objects to a sequential data set.		"Object to Sequential Data (option 6.6.2)" on page 387	OS "OS (Object to Sequential Data)" on page 1033

Table 11. File Manager panels (continued)

Option	Description	Panel	Usage	Equivalent function
6.6.3	**VSAM to Object** copies data from a VSAM data set to one or more OAM objects.		"VSAM to Object (6.6.3)" on page 387	VO "VO (VSAM to Object)" on page 1087
6.6.4	**Sequential Data to Object** copies data from a sequential data set to one or more OAM objects.	"Sequential Data to Object (option 6.6.4)" on page 387	SO "SO (Sequential Data to Object)" on page 1054	
6.6.5	**Object to Object** copies an OAM object to another object within the same collection, or to another collection.	"Object to Object (option 6.6.5)" on page 387	"Object to Object (option 6.6.5)" on page 387	OO "OO (Object to Object)" on page 1031
7	**Template and copybook utilities** allows you to create, update and edit templates and copybooks.	"Template and Copybook Utility functions panel" on page 687		
7.1	**Workbench** allows you to create, edit or update a single template.	"Template Workbench panel" on page 698	Chapter 4, "Creating and editing templates," on page 127	AF[1]
7.2	**Print** allows you to view or print copybooks or templates.	"Copybook View and Print: Entry panel" on page 507	"Copybook View and Print Utility (option 3.13 or 7.2)" on page 193	PBK "PBK (Print Copybook)" on page 1038
7.3	**Build** allows you to compile copybook(s) into template(s).	"Template Build Utility panel" on page 688	"Creating corresponding templates from selected copybooks" on page 155	BTB "BTB (Batch Template Build)" on page 841
7.4	**Update**allows you to update template(s).	"Template Update Utility panel" on page 693	"Updating one or more templates" on page 160	BTU "BTU (Batch Template Update)" on page 848
8	**Access Hierarchical File System** invokes standard ISPF services to enable access to z/OS UNIX System Services (USS) and Hierarchical File System (HFS) Utilities.	"Access Hierarchical File System panel"	Chapter 11, "Using UNIX System Services and the Hierarchical File System," on page 391	
Note: 1. This function can only be invoked under ISPF, as described in "Invoking File Manager panels from outside File Manager" on page 13.				

Access Hierarchical File System panel

The Access Hierarchical File System panel provides a means of invoking some of the z/OS Unix Services utilities.

Panel and field definitions

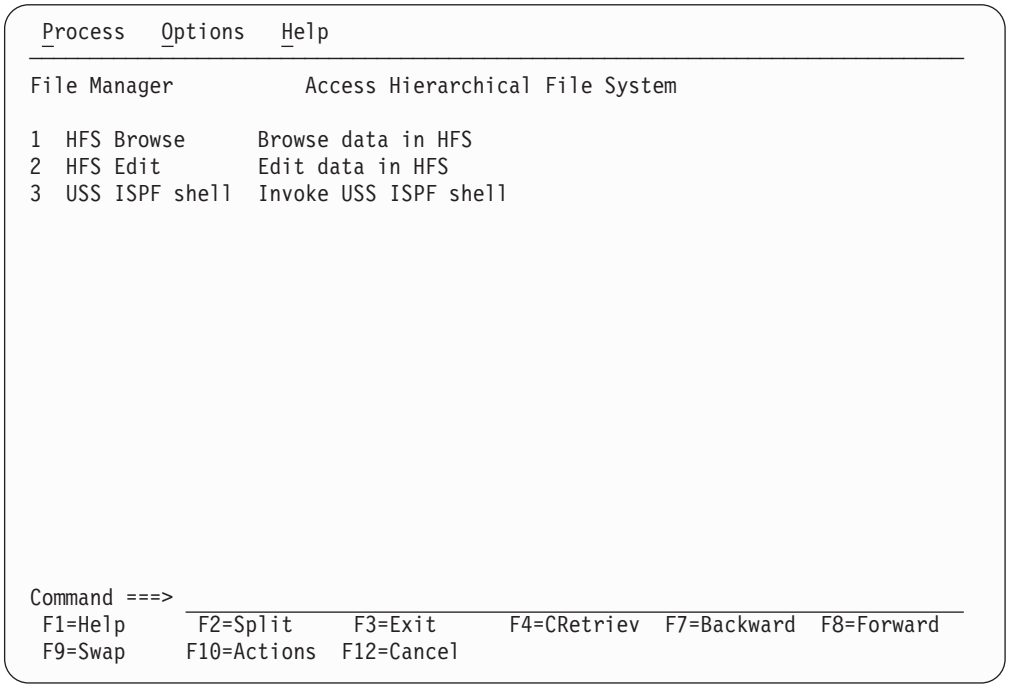


Figure 112. Access Hierarchical File System panel

1 HFS Browse
OBROWSE. Browse an HFS file using the ISPF full-screen browse facility.

2 HFS Edit
OEDIT. Edit an HFS file using the ISPF editor.

3 USS ISPF shell
ISHELL. Work with ISPF shell - a panel interface for performing various user and administrator tasks.

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

- Standard ISPF services panels.

Related tasks and examples

- Chapter 11, "Using UNIX System Services and the Hierarchical File System," on page 391

Advanced Member Selection panel

The Advanced Member Selection panel provides a means of specifying a range of PDS(E) members to be included in a print, find/change, or copy action.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager              Advanced Member Selection


Select members from:  USERID.FMDATA2

   Member name  . . . . . DATA1___     or range from: ________
   (or mask)                                  to: ________

Using ISPF Statistics criteria:

   Updated by . . . . . _______       or range from: _______
   (User id or mask)                          to: _______

   Date created . . . . . __________  or range from: __________
   (YYYY/MM/DD or mask)                       to: __________

   Date modified  . . . . __________  or range from: __________
   (YYYY/MM/DD or mask)                       to: __________


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 113. Advanced Member Selection panel

If the input data set is a load library, File Manager displays a different form of the Advanced Member Selection panel.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager  Advanced Member Selection for load libs


Select members from:  FMNUSER.LOAD


   Member name  . . . . . *_______     or range from: ________
   (or mask)                                  to: ________









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 114. Advanced Member Selection panel for load libraries

Member name (or mask)
Member name or mask to restrict the list of members eligible for processing. Cannot be specified in conjunction with a member name range.

If you have entered this field on the front panel of the utility and have selected the **Advanced member selection** option, File Manager populates this field with the same value.

or range from
The first member within a partitioned data set to be selected. Padded with blanks to 8 characters. If omitted, all members from the start of the data set to the **to** member are selected.

to The last member within a partitioned data set to be selected. Padded with blanks to 8 characters. If the last specified character is an asterisk, padded to 8 characters with high values. No other wildcarding is allowed. If omitted, all members from the **or range from** member to the end of the data set are selected.

Updated by (User id or mask)
The TSO user ID by which the member was last updated. You can specify a generic user ID by using asterisks (*) and percent signs (%) in the usual fashion.

or range from
The start of a range of user IDs. Padded with blanks to 7 characters.

to The end of a range of user IDs. Padded with blanks to 7 characters. If the last specified character is an asterisk, padded to 7 characters with high values. No other wildcarding is allowed.

Date created (YYYY/MM/DD or mask)
The date on which a member was created, in YYYY/MM/DD format.

You can use a mask to specify a combination of dates. Cannot be used in conjunction with a date range.

or range from
The start of a range of "created on" dates, in YYYY/MM/DD format. If omitted or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

to The end of a range of "created on" dates, in YYYY/MM/DD format. If omitted or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

Date modified (YYYY/MM/DD or mask)
The date on which a member was last updated, in YYYY/MM/DD format.

You can use a mask to specify a combination of dates. Cannot be used in conjunction with a date range.

or range from
The start of a range of dates on which a member was last updated, in YYYY/MM/DD format. If omitted or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

to The end of a range of dates on which a member was last updated, in YYYY/MM/DD format. If omitted or you specify an asterisk as the last character, the unspecified portions of the date default as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

Parent panels

- "Copy From panel" on page 491
- "Print Utility panel" on page 620
- "Find/Change Utility panel" on page 574

Child panels

- "Copy To panel" on page 497
- "Member Selection panel" on page 600

Related tasks and examples

- "Selecting a range of PDS(E) members" on page 34

AFP Print Browse panel

AFP Print Browse provides a formatted display of an Advanced Function Printing (LIST3820) document.

Panel and field definitions

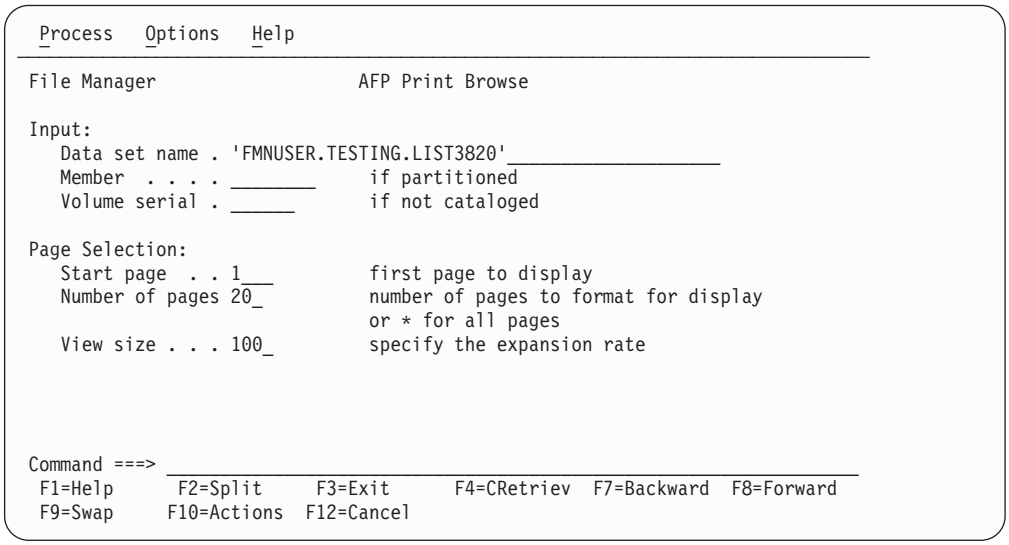


Figure 115. AFP Print Browse panel

Data set name
Can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set. Required for data sets which are not cataloged.

Start page
The number of the first page to display.

Range: 1-99999

Default: 1

Number of pages
The number of pages to display. Specify an asterisk * to include all pages to the end of the document.

Range: *,1-99999

Default: 20

View size
A measure for the virtual paper size to be used for formatting. Specify a higher value when the formatting results are not satisfactory, and a lower value to see more data on a screen.

Range: 50-400

Default: 100

Parent panels

- "Utility Functions menu panel" on page 703

Child panels

- "AFP Print Browse output"

Equivalent functions

- None.

Related tasks and examples

- "Browsing AFP Print documents" on page 329

AFP Print Browse output

AFP Print Browse output panel displays the formatted Advanced Function Printing (LIST3820) document.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                  AFP Print Browse                  Line 1 of 28
 Document FMNUSER.TESTING.LIST3820                               Col 1_______
                  Start page 1____  Number of pages 20___  View Size 100
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
=================================== Page 1 ====================================
Once a jolly swagman camped by a billabong,
Under the shade of a coolibah tree,
And he sang as he watched and waited 'til his billy boiled,
Who'll come a-waltzing, Matilda, with me?"

Waltzing Matilda, Waltzing Matilda,
Who'll come a-waltzing, Matilda, with me?
And he sang as he watched and waited 'til his billy boiled,
Who'll come a-waltzing, Matilda, with me?"

Along came a jumbuck to drink at the billabong,
Up jumped the swagman and grabbed him with glee,
And he sang as he stowed that jumbuck in his tucker bag,
"You'll come a-waltzing, Matilda, with me".
 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=NxtPage
  F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 116. AFP Print Browse: sample output

Start page
The number of the first page to display.

Range: 1-99999

Default: 1

Number of pages
The number of pages to display. Specify an asterisk * to include all pages to the end of the document.

Range: *,1-99999

Default: 20

View size

A measure for the virtual paper size to be used for formatting. Specify a higher value when the formatting results are not satisfactory, and a lower value to see more data on a screen.

Range: 50-400

Default: 100

Available commands

The following primary commands are available on this panel:
- FIND - see "FIND/FX primary command" on page 769
- LOCATE - see "LOCATE primary command" on page 786
- NEXTPAGE- this command is unique to the AFP Browse panel.
- PREVPAGE- this command is unique to the AFP Browse panel.

The NEXTPAGE primary command positions the display down to the beginning of the next page.

The PREVPAGE primary command positions the display up to the beginning of the previous page.

Parent panels

- "AFP Print Browse panel" on page 442

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Browsing AFP Print documents" on page 329

AIX Association Information panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager          AIX Association Information

AIX Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1'
   Catalog ID . . . . 'CATALOG.SYSPLEXD.USER'
                                                                      More:
AIX Associations:
   Cluster  . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8'
     Data component . 'SYATES.UNITTEST.RFM0189.KSDS8.DATA'
     Index component  'SYATES.UNITTEST.RFM0189.KSDS8.INDEX'
   Path . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1PTH1'
   Path . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1PTH2'
                      ****  End of data  ****




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Info      F7=Up
 F8=Down      F9=Swap     F10=Actions  F12=Cancel
```

Figure 117. AIX Association Information panel

Parent panels

- "AIX Entry Detail panel" on page 447

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Viewing association information" on page 316

AIX Entry Detail panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  AIX Entry Detail

AIX Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.ESDS2.AIX1'
   Catalog ID . . . . 'CATALOG.UCATAPC'
                                                                        More:     +
Basic Information:
   Creation date  . . 2003.351         Expiration date (NONE)

   Data component . . 'SYATES.UNITTEST.RFM0189.ESDS2.AIX1.DATA'
   Index component  . 'SYATES.UNITTEST.RFM0189.ESDS2.AIX1.INDEX'

AIX Associations:
   Cluster  . . . . . 'SYATES.UNITTEST.RFM0189.ESDS2'
     Data component . 'SYATES.UNITTEST.RFM0189.ESDS2.DATA'
   Path . . . . . . . 'SYATES.UNITTEST.RFM0189.ESDS2.AIX1PTH1'

SMS Attribute:
   SMS managed  . . . Y
AIX Attributes:
   CI size  . . . . . 20480    size of the data control intervals
   Buffer space . . . 41984    buffer space to be allocated at open time

   Share options .    Cross region  1           Cross systems 3

   Process options    Reuse . . . . N           Recovery  . . Y
                      Erase . . . . N           Extended  . . N
                      Upgrade . . . Y           Writecheck  . N

 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Assocs    F7=Up
 F8=Down      F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 118. (Part 1 of 2) AIX Entry Detail panel

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  AIX Entry Detail

AIX Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.ESDS2.AIX1'
   Catalog ID . . . . 'CATALOG.UCATAPC'
                                                                    More:     -
AIX Data Allocation:
   Allocation unit  . CYL      REC, KB, MB, TRK, or CYL

   Space . . . . .    Primary . . 1            Secondary . 0
   Record size . .    Average . . 4086         Maximum . . 32600
   Free space . .     % of CI . . 0            % of CA . . 0

   Volume serial(s) . MV8W17
   Device type(s) . . 3390

AIX Key Definition:
   Key length . . . . 64                  Key offset  . . 0
   Unique . . . . . . N

AIX Index Allocation:
   CI size  . . . . . 1024     size of the index control intervals
   Allocation unit  . TRK      REC, KB, MB, TRK, or CYL

   Space . . . . .    Primary . . 1            Secondary . 0

   Volume serial(s) . MV8W17
   Device type(s) . . 3390
Command ===>
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Assocs    F7=Up
  F8=Down      F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 119. (Part 2 of 2) AIX Entry Detail panel

Available commands

- "ASSOCS primary command" on page 741
- "STATS primary command" on page 822

Parent panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Child panels

- "AIX Association Information panel" on page 446

Related tasks and examples

- "Viewing your Catalog Entry Details" on page 316

Allocate panel

You use the Allocation panel to select the data set organization type for a new data set. File Manager displays this panel whenever you specify an output data set that does not exist, for example, as part of the Data Create or Copy process.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
Allocate /u/testonly/xxx.txt

New Data Set Organization:
   Select option         Instructions
   _ 1. KSDS             The above data set does not exist.
     2. ESDS             To define or allocate a new data set select a data
     3. RRDS             set organization and press ENTER or press PF3/EXIT
     4. VRRDS            or PF12/CANCEL to return without allocation.
     5. LDS
     6. Non VSAM
     7. IAM KSDS         For a new data set, enter a data set name
     8. IAM ESDS         below to copy existing allocation attributes.
     9. HFS

Existing Data Set:
   Like data set . . . . . ____________________________________________
   Volume serial . . . . . ______


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 120. Allocation panel

Allocate
Lists the new data set name that you are allocating.

New Data Set Organization
Option to select the data set organization type. Choose from:

1. KSDS
Key-sequenced data set. A type of VSAM data set that contains records in ascending collating sequence, and can be accessed by a field, called a key, or by a relative byte address.

2. ESDS
Entry-sequenced data set. A type of VSAM data set that contains records in the order in which they were entered. Records are added to the end of the data set, and can be accessed.

3. RRDS
Relative record data set. A type of VSAM data set that contains records in order by relative record number, and can be accessed only by this number. There are two types of RRDS: fixed-length and variable length. In this panel, the term RRDS refers specifically to the fixed-length type. However, in other situations where the types do not need to be differentiated, the term RRDS may be used for both types.

4. VRRDS
Variable length relative record data set. A type of VSAM data set that contains records in order by relative record number, and can be accessed only by this number. There are two types of RRDS: fixed-length and variable length. Within File Manager, the term VRRDS refers specifically to the variable-length type.

5. LDS
Linear data set. A type of VSAM data set that contains data that has no record boundaries. Linear data sets contain none of the control information that other VSAM data sets do and must be cataloged.

6. Non-VSAM
Allows you to specify a sequential or partitioned data set, using either an existing data set as a model, SMS class names and disk space requirements, or the system defaults.

7. IAM KSDS
Indexed Access Method - Key Sequence. A non-IBM data set type.

8. IAM ESDS
Indexed Access Method - Entry Sequence. A non-IBM data set type.

9. HFS
Hierarchical File System file. A non-z/OS file, processed in File Manager as a simulated QSAM/BSAM data set.

Note: File Manager only displays this option when you have specified an HFS path name (that is, a name starting with a "/") in the **Data set/path name** field of the preceding panel.

Like data set
The name of an existing sequential or partitioned data set to be used as a model.

If you specify a mask in this field, File Manager displays the Model Data Set Selection panel listing the data sets matching the mask from which you can select the model data set.

The attributes of this data set are copied to the second allocation panel as the default values for your new data set.

Volume serial
The volume serial of an existing non-cataloged sequential or partitioned data set to be used as a model when option **6. Non-VSAM** has been selected.

You can only specify a volume serial if you have also specified a model data set (or mask) in the **Like data set** field.

Parent panels

- "Data Create Utility panel" on page 515
- "Copy To panel" on page 497

Child panels

- "VSAM Define panel" on page 719
- "Allocate (Two) panel" on page 451
- "IAM KSDS Define panel" on page 582

Equivalent functions

- None.

Related tasks and examples

"Allocating a new data set" on page 246

Allocate (Two) panel

This is the second Allocate panel, in which you can define the attributes of a non-VSAM data set.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
Allocate USERID.NEWDS

Specify a model data set, SMS class names, or leave blank for defaults:
   Like data set . . ____________________________________________
   Volume serial . . ______
   Device type . . . _____        generic unit or device address
   Data class  . . . ________     leave blank for default
   Storage class . . ________     leave blank for default
   Management class  ________     leave blank for default

Space Requirements:
   Space unit  . . . TRK          BLKS, TRKS, CYLS, KB, or MB
   Primary units . . 5            quantity of above units
   Secondary units . 5            quantity of above units
   Directory blocks  ______       leave blank for SMS default
   Record format . . FB           if new format: U,F,V, or D, with B,S,A,M
   Record length . . 50
   Block size  . . . _____        physical output block size
   Library type  . . _______      LIBRARY, PDS, or blank for default
   Release unused  . NO           enter YES to free unused space, else NO
   Expiration date . ________     yyyy.ddd, blank for default
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 121. Allocate (Two) panel

Like data set
The name of an existing data set, to be used as a model for your new data set.

Volume serial
The volser of an existing data set, to be used as a model for your new data set.

Device type
Generic unit (EDT Esoteric Device Type) or device address. This field is mutually exclusive with the **Volume serial** field.

Data class
The name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage class
The name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Management class
The name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Space unit
Defines the unit of primary and secondary space to be allocated. A unit type is required, when a quantity is specified. Can be one of:

BLK Block of average size.

KB Kilobyte (1024 bytes).

MB Megabyte (1048576 bytes).

TRK Track of a direct access storage device (DASD).

CYL Cylinder of a DASD.

Primary units
Amount of DASD space to be used for primary space allocation. The range depends on the space unit specified and the DASD device type.

Secondary units
Amount of DASD space to be used for secondary space allocations. The range depends on the space unit specified and the DASD device type.

Directory blocks
Number of blocks for the directory. You must specify a value when on a system without SMS, or when your SMS installation does not provide a default.

Range: 0-999999

Record format
Enter the record format, optionally followed by one or more attribute characters in the sequence shown. Available formats:

F Fixed

V Variable

D ASCII variable (tape only)

U Undefined

Attributes:

B Blocked

S Spanned, V and D format only

A with ANSI control characters

M with machine code control characters.

If omitted, the record format of an old data set is not changed. For a new data set, the input record format (if any), or U is used.

Note: To create data with a record format not shown here, allocate the data set with the TSO ALLOCATE command to ddname QSAMOUT.

Record length
The length of the records to be created. Range is dependent upon the record format.

If the record format is VBS, the record length may be specified as 32768, meaning that you are defining an LRECL=X data set.

Block size
The average size of the physical blocks to be written.

Range: 0-32760

Library type
Library or **PDS** or blank for the system default. For type **Library**, an active SMS is required.

Release unused
Specify YES to release any unused space when the data set is closed. The default is to keep the space allocated.

Parent panels

- "Allocate panel" on page 448

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

"Allocating a new data set" on page 246

Allocate (Three) panel

This is the third Allocate panel, in which you can define the attributes of an HFS file.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
Allocate /u/testonly/xxx.txt

Specify attributes of a HFS file:

Select access level (0-7) for each class of users:
   Owner  . . . . 0                    0 - No access
   Group  . . . . 0                    1 - Search access
   Other  . . . . 0                    2 - Write-only access
                                       3 - Write and search access
                                       4 - Read-only access
                                       5 - Read and search access
                                       6 - Read and write access
                                       7 - Read, write, and search access









Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 122. Allocate (Three) panel

Owner
Defines the access level to the created file for the owner. The access is specified by a number 0–7 (default 7). See below for details.

Group Defines the access level to the created file for the group. The access is specified by a number 0–7 (default 7). See below for details.

Other Defines the access level to the created file for other users. The access is specified by a number 0–7 (default 5). See below for details.

For each of the above classes of user, you specify the access level as 1–7:

0 No access

1 Search access

2 Write-only access

3 Write and search access

4 Read-only access

5 Read and search access

6 Read and write access

7 Read, write, and search access

Parent panels

- "Allocate panel" on page 448

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Allocating a new data set" on page 246
- "Creating HFS files" on page 394

Append panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                        Append

To Data set:
   Data set/path name  . . 'FMN.FMDATA'
   Member name (or mask) . ________
   Volume serial . . . . . ______

Allocation Options:
1  1. Allocate using the attributes of
        'FMNUSER.DATA'
   2. Specify allocation attributes

Processing Options:
ISPF Packing                         Enter anything to select option
1  1. None                           _  Binary mode, reclen _____
   2. Pack



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 123. Append panel

The **Allocation Options** are only required when the "To" data set does not exist.

The **Binary mode** and **reclen** parameters are only relevant for an HFS file.

Parent panels

Child panels

- "Allocate panel" on page 448
- "Allocate (Two) panel" on page 451
- "Allocate (Three) panel" on page 453

Equivalent functions

- None.

Related tasks and examples

- "Creating, replacing, and adding data using existing data" on page 246

Browse panel

You use the Browse panel to display a selected data set or data set member, scroll through the records and find specific information within the records.

The Browse panel displays different fields in the header area, depending upon the type of data set shown and whether or not a template has been used.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
Browse           FMN.V11R1M0.SFMNSAM1(FMNCDATA)                Rec 0 of 40
                                                      Record 0        Format TABL
REC-TYPE REC-ID NAME                 EMPLOYEE-NO      AGE   SALARY    MONTH(1)
#2       #3 R # #4                             #5      #6       #7          #8
AN 1:2   AN 1:2 AN 3:20                  BI 23:2 BI 25:2  PD 27:4     BI 31:4
<>       <>     <---+----1----+---->      <---+>  <---+> <---+--> <---+----1>
****  Top of data  ****
01       01     Grant Smith                 7712      35    75000           6
01       01     Andrew Apple                6645      53    78500          30
01       01     Graham Prestcott            5583      28    48000           7
01       01     Bill Somers                 4418      33    68000           5
01       01     Ted Dexter                  3327      52    60250          14
01       01     Roddy Armstrong             5683      34    77000          28
01       01     Cliff Roberts               2265      57   100000          44
01       01     James Browne                1117      46   125000          47
01       01     Silvia Carrot               2308      29    61400           1
01       01     Dan Peters                  4479      54    63000          38
01       01     John Laws                   3422      21    46750          14
01       01     Liz Childs                  3439      55    66000          44
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 124. Browse panel showing text file

Title The Title identifies the function (Browse) and the data set being used. For a PDS or PDSE member the data set name includes the member name. Record number information, or short messages, appear to the right.

Record
The Record field shows the record number of the record that is currently at the top of the data area. You can scroll to a specific record by entering a new value in this field.

Note: The number shown is that of the record within the entire data set, regardless of any restrictions, such as selection criteria, that you may have used to limit the display of records.

The default value for this field is 0, which positions the display before the first record. When the field contains 0, the data area displays the

```
****  Top of data  ****
```

message. Once you have scrolled away from this position, you cannot return to it by typing 0 in the Record field. Instead you need to enter a scrolling command, such as TOP, to return to 0.

Col The Col field shows the column number of the column that is currently at the far left of the data area. You can scroll to a specific column (left or right) by typing a new value. The default value for this field is 1.

Scale The Scale shows the columns of the data area.

Data Area
The Data Area shows the data in the selected display format. For a description of the different display formats, see "Selecting a display format" on page 66.

Command
The Command line is a field in which you can enter Primary Commands, such as FIND.

Scroll The Scroll field defines the current scroll amount. You can type a new value.

When a VSAM data set is displayed, additional fields can be seen.

```
Browse           FMN.REQ77.RRDS                                    Rec 0
                          Type RRDS Slot __________ RBA           Format CHAR
       Col 1
RBA         Len    ----+----1----+----2----+----3----+----4----+----5----+----
80           80    01Tyrone Dallas       .0...-.............................
```

Figure 125. Browse panel showing VSAM data set

Type The type of VSAM data set, for example, RRDS. IAM files are also indicated.

Slot The slot number of the currently selected RRDS file. You can type a new value in this field to move to a new file.

RBA The Relative Byte Address of the currently selected RRDS file.

Column Headings
The Scale shows the column headings for the RBA (Relative Byte Address) and Len (Record Length) of each RRDS file in the data set.

When the VSAM data set is a KSDS file, the Key field can be used to position yourself within the data set.

```
Browse           FMN.REQ77.KSDS                                      At top
       Type KSDS     Key ____________________________ RBA ___________ Format CHAR
       Col 1
RBA         Len    <===+====1====+====2====+====3====+====4====+====5====+====
****  Top of data  ****
0            64    00000010AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAb
64           64    00000020AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Figure 126. VSAM KSDS data set

Key If the value you enter contains leading, embedded or trailing blanks, commas or quotation marks, it must be enclosed in quotation marks. You may also enter a hexadecimal string enclosed in quotation marks and preceded by X or x, for example, `X'C1C2'`. The maximum number of characters, including any required characters, is 250.

When your cursor is in this field, you can use the LEFT/RIGHT commands to scroll within the field (function keys F10/F11). You can also use the EXPAND function (F4), to open the Key field in a window.

When a data set is displayed in TABL format with a template, the column headings show the field names defined in the template.

```
Browse          FMN.V11R1M0.SFMNSAM1(FMNCDATA)                Rec 0 of 40
                                                  Record 1        Format TABL
REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY  MONTH(1)  MONTH(2)
#2       #3                           #4      #5       #6        #7        #7
AN 1:2   AN 3:20                 BI 23:2 BI 25:2  PD 27:4   BI 31:4   BI 35:4
<>       <---+----1----+---->      <---+>  <---+> <---+--> <---+---> <---+--->
01       Grant Smith                7712      96    75000         6        15
```

Figure 127. PDSE data set member in TABL format

When a data set is displayed in SNGL format with a template, the field and record number of the current record is shown.

```
Browse          FMN.V11R1M0.SFMNSAM1(FMNCDATA)                Rec 21 of 40
Current type is REC-TYPE02                                        Format SNGL
                                 Top Line is 1    of 7     in Record 21
Ref Field    Picture Typ Start   Len   Data
  2 REC-TYPE
              XX      AN      1     2   02
  3 NAME      X(20)   AN      3    20   Grant Smith
  5 ADDR1     X(20)   AN     37    20   22  Montrose St
  6 ADDR2     X(20)   AN     57    20   Thornlie
  7 POSTCODE
              X(4)    AN     77     4   6145
****  End of record  ****
```

Figure 128. PDSE data set in SNGL format

Available commands

- "BOTTOM primary command" on page 742
- "CANCEL primary command" on page 744
- "CEDIT primary command" on page 747
- "DEDIT primary command" on page 758
- "DOWN primary command" on page 760
- "END primary command" on page 762
- "EXCLUDE/XX primary command" on page 762
- "EXIT primary command" on page 766
- "FE (Find Error) primary command" on page 767
- "FILE primary command" on page 769
- "FIND/FX primary command" on page 769
- "FORMAT primary command" on page 780
- "HEX primary command" on page 780
- "JUST primary command" on page 782
- "LEFT primary command" on page 784
- "LOCATE primary command" on page 786
- "NEXT primary command" on page 793
- "NEXTREC primary command" on page 794
- "OFFSET primary command" on page 795
- "PIC primary command" on page 797
- "PREVIOUS primary command" on page 798
- "PREVREC primary command" on page 799
- "RBALEN primary command" on page 800
- "RECLEN primary command" on page 803
- "PROFILE primary command" on page 800
- "RD primary command" on page 801
- "RDF primary command" on page 802
- "RECSTATS primary command" on page 804
- "REFS primary command" on page 805

- "RESET primary command" on page 806
- "RFIND primary command" on page 809
- "RIGHT primary command" on page 810
- "RP primary command" on page 811
- "SHADOW primary command" on page 816
- "SHOW primary command" on page 817
- "SLOC primary command" on page 818
- "SORT primary command" on page 818
- "STR primary command" on page 821
- "TEDIT primary command" on page 824
- "TOP primary command" on page 824
- "TVIEW primary command" on page 825
- "TYPE primary command" on page 826
- "UP primary command" on page 826
- "VIEW primary command" on page 828
- "ZOOM primary command" on page 829

Parent panels

- "Browse Entry panel"

Child panels

From the Browse panel, you can use primary commands to access the following panels:

- "Template Workbench panel" on page 698 (TVIEW).
- "Record Type Selection panel" on page 636 (TEDIT - panel displays when template contains more than one record type).
- "Field Selection/Edit panel" on page 565 (TEDIT - panel displays when using a copybook template with only one record type).
- "Dynamic Template panel" on page 529 (TEDIT - panel displays when using a dynamic template).
- "Record Selection Criteria panel" on page 631 (CEDIT).
- "Record Identification Criteria panel" on page 625 (CEDIT ID).

Equivalent functions

- "DSB (Data Set Browse)" on page 868

Related tasks and examples

- Chapter 3, "Viewing and changing data sets," on page 49
- "Manipulating your view of the data" on page 64
- "Viewing segmented data" on page 229
- "Segmented data templates" on page 131

Browse Entry panel

You use the View Entry panel to select a data set for viewing in the Browse panel.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Browse Entry Panel

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name 'MACHIND.SEQ1M'                                  +
   Member . . . . . . ________     Blank or pattern for member list
   Volume serial  . . ______       If not cataloged
   Start position . . ____________________________  +
   Record limit . . . ________     Record sampling  _

Copybook or Template:
   Data set name  . . 'FMN.TEMPLATE'
   Member . . . . . . TEST01       Blank or pattern for member list
Processing Options:
 Copybook/template   Start position type   Enter "/" to select option
 3  1. Above          _  1. Key             _  Edit template   Type (1,2,S)
    2. Previous       _  2. RBA             _  Include only selected records
    3. None           _  3. Record number   _  Binary mode, reclen _____
    4. Create dynamic    4. Formatted key

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 129. Browse Entry panel

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set. Required for data sets which are not cataloged.

Start position
Initial starting position for the data set to be browsed. The initial display is positioned at the specified record.

The default is the top of the data set. You can enter a negative record to indicate the number of records before the end of file. For example, to see just the last record on the file, enter -1 as the start point.

The format of the start position field is either numeric or character, depending upon the type of start position selected. For VSAM KSDS Key values, if the value you enter contains leading, embedded or trailing blanks, commas or quotation marks, it must be enclosed in quotation marks. You may also enter a hexadecimal string enclosed in quotation marks and preceded by X or x, for example, `X'C1C2'`. The maximum number of characters, including any required characters, is 250.

When your cursor is in this field, you can use the LEFT/RIGHT commands to scroll within the field (function keys F10/F11). You can also use the EXPAND function (F4), to open the key field in a window.

Note: The Erase EOF key only works on the displayed part of the key. When the key being displayed is larger than the field area on the screen, you must either scroll or expand the field to erase the unseen portions.

For all other data set formats, a valid unsigned number must be entered.

To specify a starting position as the number of records before the end of file, enter a negative record number. For example, to see just the last record on the file, enter a start position of -1.

If you specify a starting position, you should also specify the **Start position type** in the Processing Options. If you do not, File Manager assumes that the value provided in the Starting position field is a record number.

VSAM - KSDS: Key value
If the key is not matched, the record with a key greater than the value given is specified. If the key value is greater than the last record in the data set, the initial position is the “End of data” marker.

VSAM - ESDS: RBA (Relative Byte Address) Value
If the RBA is not matched, the record with a RBA greater than the value given is specified. If the RBA value is greater than that of the last record in the data set, the initial position is the “End of data” marker.

VSAM - RRDS: Slot value
If the slot number is greater than the last used slot in the data set, the initial position is the “End of data” marker.

QSAM: Record Number
If the Record Number is greater than that of the last record in the data set, the initial position is the “End of data” marker.

HFS As for QSAM.

Record limit
This field restricts the number of records retrieved from a data set (from the start point or top) resulting in an edit or browse of a portion of the data set. You can use the keyword “MEMORY” as a record limit to restrict the number of records retrieved to as many as will fit comfortably in the available virtual storage.

The default start position is the top of the data set.

Record Sampling
Indicates whether you want record sampling to be performed on the data set. If you select this option, File Manager displays the Record Sampling panel.

Copybook or Template
Data set name and Member name of the template or copybook to be used to format your data.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Copybook/template usage
Indicates if you want to use a template for a logical view of the data.

1 Use the template specified on the panel (or compile the specified copybook into a template, and use that).

2 Use the last template associated with the data set.

3 No logical view is to be used by the function.

4 Create a dynamic template.

Start position type
Determines how File Manager interprets the value provided in the Starting position field. If the Start position type is not specified, File Manager assumes that the value is a Record Number.

1. KEY
Only valid when the data set is a VSAM KSDS, VSAM AIX or VSAM PATH.

2. RBA
Only valid when the data set is a VSAM KSDS, VSAM AIX or VSAM ESDS.

3. Record Number
Default. Valid for any type of supported data set.

4. Formatted key
Only valid when the dataset is a VSAM KSDS, VSAM AIX, or VSAM PATH and when a template is provided. This displays the formatted key positioning panel.

Note: For templates with more than one layout, a 01 selection list is displayed first.

Edit template
Indicates if you want to edit the template before use. You edit the template when you need to change format, selection, data create attributes, and reformatting information for output copy templates.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Include only selected records
When the Edit or Browse session is either in-storage (a record limit of "MEMORY" is used, or omitted and set on by means of this option) or

when you have selected **Record Sampling**, only the records selected by template processing are included in the Edit or Browse session.

This option does not alter the operation of edit or browse if a template is not used.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The fields only display if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. I/O exits can only be used to process the data sets that you are using. They cannot be used to process the copybook or template that you are using to format the data set.

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

Depending upon the options selected in the Browse Entry panel, the next panel displayed can be:

- "Browse panel" on page 455
- "Data Set Selection panel" on page 519 (a pattern has been entered in the Data set name field)
- "Member Selection panel" on page 600(a pattern or a blank has been entered in the Member field)
- "Record Type Selection panel" on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- "Field Selection/Edit panel" on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).
- "Dynamic Template panel" on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- "Record Sampling panel" on page 628 (Record Sampling option selected)
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).

- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

- "DSB (Data Set Browse)" on page 868

Related tasks and examples

- "Supplying a procedure when using a File Manager panel" on page 417
- "Starting an editor session without using templates" on page 51

Catalog Services panel

The Catalog Services panel is used to list or print catalog entries, and to invoke IDCAMS commands or other File Manager options for catalog entries.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Catalog Services

  blank List catalog entries               A Alter catalog entry
    DEF Define catalog entry             DEL Delete catalog entry
      I Display entry information          P Print catalog entries

Data Set:
   Data set name . 'FMNUSER.EXPORT'
   Catalog ID  . .

Processing Options:
   Entry Type
   3  1. Any           5. Alias         8. GDG          11. Page space
      2. Non-VSAM      6. Cluster       9. Index        12. Path
      3. VSAM          7. Data         10. OAM          13. User catalog
      4. AIX

   Enter "/" to select option
   _  Batch execution  /  with list
      Include Additional Qualifiers
   7  YY/MM/DD date format (def. YYYY.DDD)
   _  Processing limit  0
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 130. Catalog Services panel

Data Set

Describes the data set to be processed:

Data set name

The data set name to be used as the target of the catalog service you have requested. For the list and print services, you can specify a generic data set name.

Catalog ID

The catalog to be searched in place of using the system catalog search order.

Processing Options

Entry type
Restricts the catalog search to the specified type. The effect of this option depends on the catalog service that has been requested. For all services except DEFINE, it simply restricts the search to entries of the specified type and is useful with the list or print service and a generic data set specification. With the DEFINE service, it predetermines or restricts the type of entry to be defined. If the type is ambiguous (Any, VSAM, Cluster, Data, or Index), File Manager displays a pop-up panel from which you can specify a specific entry type to be defined.

Batch execution
Presents the JCL to run the LIST, PRINT, DEFINE, or DELETE functions in batch. You can edit the JCL before submitting it. Use the Set Batch Job Card Information panel to tailor the default JOB card that File Manager uses to generate JCL.

with list
Used in conjunction with the **Batch execution** option. File Manager first displays a data set list matching the entered data set name allowing you to specify DEFINE or DELETE functions using prefix commands to produce JCL for a data set selected from the list.

Include Additional Qualifiers
When selected, generates the data set list with all data sets matching the qualifiers in the **Data set name** field, including data sets with additional qualifiers.

When not selected, the data set list is restricted to only include data sets with the qualifiers entered in the **Data set name field**. Data sets with additional qualifiers are not included.

YY/MM/DD date format (def. YYYY.DDD)
When selected, causes all dates on the Data Set List panel and listing to be shown as YY/MM/DD instead of the default YYYY.DDD.

Processing limit
When selected, only the specified number of entries are processed for display or print.

Parent panels

- "Utility Functions menu panel" on page 703

Child panels

- "AIX Entry Detail panel" on page 447
- "Catalog Services Data Set List panel" on page 466
- "Delete Entry panel" on page 520
- "GDG Entry Detail panel" on page 581
- "IAM KSDS Define panel" on page 582
- "IAM Entry Details panel" on page 583
- "Non-VSAM Entry Detail panel" on page 609
- "Path Entry Detail panel" on page 612
- "VSAM Define panel" on page 719
- "VSAM Entry Detail panel" on page 722

Equivalent functions

- "SCS (Catalog Services)" on page 1045

Related tasks and examples

- "Managing catalog entries" on page 311

Catalog Services Data Set List panel

Panel and field definitions

```
 Process   Options   Help
 File Manager                 Data Set List                Row 00001 of 00027
Catalog ID ''                                                    Types ALL
        Data Set Name                                    Type  Volume  MV Creat ±
        FMN.V11R1M0.S*                                   *     *       *  *
_______ FMN.V11R1M0.SFMNCLIB                             NVSAM D$FM08     09/12
_______ FMN.V11R1M0.SFMNDBRM                             NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNEXEC                             NVSAM D$FM10     09/12
_______ FMN.V11R1M0.SFMNMAC1                             NVSAM D$FM00     09/12
_______ FMN.V11R1M0.SFMNMENU                             NVSAM D$FM19     09/12
_______ FMN.V11R1M0.SFMNMJPN                             NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNMKOR                             NVSAM D$FM03     09/12
_______ FMN.V11R1M0.SFMNMODA                             NVSAM D$FM02     09/12
_______ FMN.V11R1M0.SFMNMODJ                             NVSAM D$FM00     09/12
_______ FMN.V11R1M0.SFMNMODK                             NVSAM D$FM10     09/12
_______ FMN.V11R1M0.SFMNMOD1                             NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNMOD2                             NVSAM D$FM04     09/12
_______ FMN.V11R1M0.SFMNPENU                             NVSAM D$FM09     09/12
_______ FMN.V11R1M0.SFMNPJPN                             NVSAM D$FM16     09/12
_______ FMN.V11R1M0.SFMNPKOR                             NVSAM D$FM13     09/12
Command ===>                                                   Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 131. Catalog Services: list of catalog entries (initial display)

Available commands

In addition to the commands listed in "Manipulating your view of the data" on page 64, the following primary command is available on this panel:

- "AMSMSG primary command" on page 739

The line commands you can use with a list of catalog entries are:

AFP Displays AFP (LIST3820) data using the File Manager AFP utility.

ALTER (A)
Lets you alter the parameters for the selected entry. A panel is displayed on which you can change parameters for the entry, as described in "Managing catalog entries" on page 311.

To run the ALTER command in batch, first select the **Batch execution** option on the Catalog Services panel

BROWSE (B)
If the selected entry is an OAM object collection or a catalog, invokes the appropriate File Manager list option (Object Directory List or Catalog Services); otherwise, invokes Browse.

COPY (C)
Copies the selected entry. Invokes the Copy Utility (3.3) with the "from" data set prefilled to the selected data set.

DEFINE (DEF)
Lets you define a new entry that is similar to the selected entry. A panel is displayed on which you enter the name of the new entry. You can also change the parameters (copied from the existing entry), as described in "Managing catalog entries" on page 311.

To run the DEFINE command in batch, first select the **Batch execution** option on the Catalog Services panel

DELETE (DEL)
Deletes the selected entry.

To run the DELETE command in batch, first select the **Batch execution** option on the Catalog Services panel

EDIT (ED)
Invokes Edit (option 2).

EXTENTS (EX)
For a data set entry, lists the statistics and extent information in the catalog.

INFO (I)
Displays the Non-VSAM Entry Detail panel or VSAM Entry Detail panel. Synonymous with **LIST**. If entered for a catalog, reinvokes catalog services to list the contents of the catalog. For all other catalog entries, displays the details and, for VSAM entries, also statistics and allocation information. The information displayed is the same as would be displayed for the selected entry using the information service from the entry panel.

LIST (L)
If the selected entry is a catalog, invokes Catalog Services for that catalog; otherwise, lists the parameters for the selected entry. Synonymous with **INFO**.

MEMBER (M)
If entered for a library data set entry, displays a member list.

PRINT (P)
Invokes Print Utility (option 3.2).

RECALL (REC)
If the selected entry has been migrated or archived, this command recalls the data set.

RENAME (R)
This is a synonym for Alter with non-VSAM data sets and it works identically. For VSAM data sets it invokes a separate VSAM Entry Rename panel.

VIEW (V)
Invokes AFP Print Browse (option 3.7) for non-VSAM data sets.

Note: Only use this command for data sets that contain data in Advanced Function Printing (LIST3820) format.

Line commands work differently according to the type of catalog entry. Table 12 on page 468 shows the line commands that are available.

Table 12. File Manager line commands available for different types of entries

	Alter	Define	Delete	List	Browse	Edit	Print	View	Extents	Info
AIX	yes	yes	yes	yes			yes		yes	yes
Alias		yes	yes	yes						
Cluster[3]	yes	yes	yes[4]	yes	yes	yes	yes		yes	yes
GDG		yes	yes	yes						
Non-VSAM	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
OAM				yes	yes[5]					
Path	yes	yes	yes	yes	yes[6]	yes[6]	yes			
PGSPC				yes					yes	yes
UCAT				yes[7]					yes	yes

Note:
1. A blank space means that the line command is not available.
2. The word "yes" means that the line command is available.
3. Represents the types: CLUST, DATA, ESDS, INDEX, KSDS, LDS, RRDS, and VRRDS. "CLUST" is only shown for a base component that has a DATA or INDEX component catalog error.
4. You can delete a cluster, but the deletion attempt might fail if the cluster is large or complicated (for example, with more than 1 AIX or more than 20 volsers per component).
5. Object Directory List (option 6.1).
6. Is only successful if the path is defined over a base cluster rather than an alternate index.
7. Catalog Services is invoked again to list the entries in a catalog.

Parent panels

- "AIX Entry Detail panel" on page 447
- "Delete Entry panel" on page 520
- "GDG Entry Detail panel" on page 581
- "IAM KSDS Define panel" on page 582
- "IAM Entry Details panel" on page 583
- "Non-VSAM Entry Detail panel" on page 609
- "Path Entry Detail panel" on page 612
- "VSAM Define panel" on page 719
- "VSAM Entry Detail panel" on page 722

Equivalent functions

- "SCS (Catalog Services)" on page 1045

Related tasks and examples

- "Managing catalog entries" on page 311
- "Manipulating your view of selection lists" on page 28

Clipboard Manager panel

The Clipboard Manager panel lets you see and modify the clipboards available to the PASTE primary command. Clipboards are created using the CUT primary command in edit.

Panel and field definitions

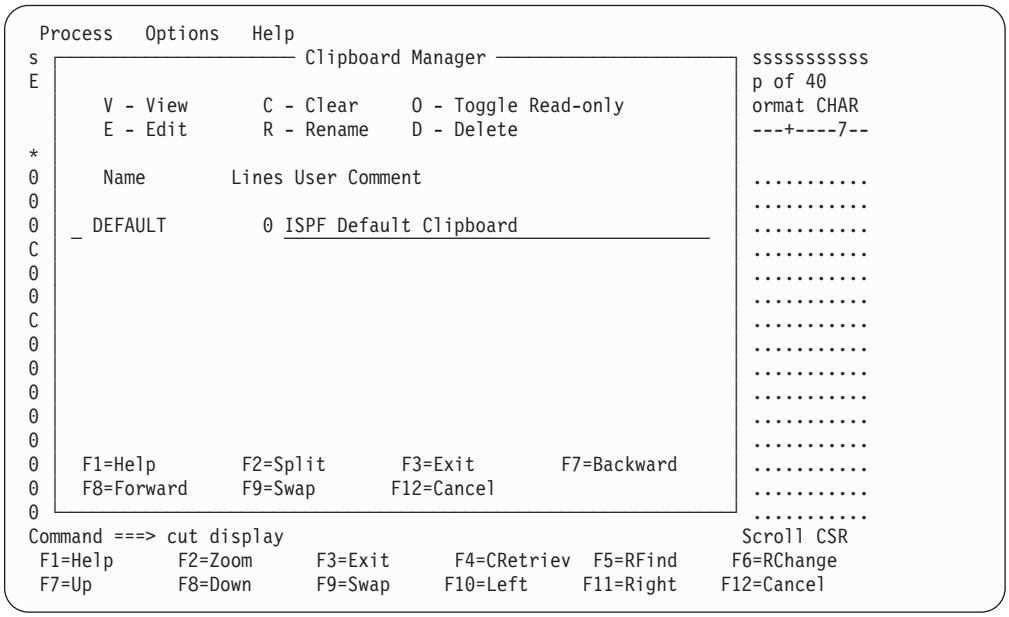


Figure 132. Clipboard Manager panel

Line command entry field
You can enter these line commands to the left of the clipboard name:

B Browse the contents of the clipboard using ISPF browse.

C Empty the contents of the clipboard. You can also replace the contents of a clipboard by using the REPLACE option of the edit CUT command.

D Delete a clipboard.

E Edit a clipboard in the ISPF editor.

O Make a clipboard read only, or change a read-only clipboard to modifiable. When a clipboard is read-only, the CUT command does not modify the contents of the clipboard. This is useful when you are going to PASTE the clipboard contents into several places and you want to be sure the data is not accidentally overwritten by a CUT command. The comment field can not be changed for read-only clipboards.

R Rename a clipboard

Lines The number of lines in the clipboard.

User Comment
The name of the clipboard.

Parent panels

- "Editor panel" on page 534

Child panels

None

Related tasks and examples

- "Copying data to and from a clipboard" on page 121

Compiler Language Selection panel

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager             Compiler Language Selection

Language Selection:
   1  1. COBOL          Use the COBOL compiler
      2. PL/I           Use the PL/I compiler
      3. Auto detect    Determine which compiler to use.
      4. HLASM          Use the HLASM compiler

Processing Options:
   Enter "/" to select option
   _ Override compiler options for template update
   _ Preserve copybook library
   _ Show copybook name with record type field name
   _



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 133. Compiler Language Selection

Language Selection
The language used for compiling a copybook to create a template.

COBOL
Use the COBOL compiler. This is the default.

PL/I Use the PL/I compiler.

Auto detect
Analyze the source to determine whether the language of the source is COBOL or PL/I to use the appropriate compiler

Note: Selecting Auto detect may increase processing time of your copybooks.

HLASM
Use the HLASM compiler.

Processing Options

Override compiler options for template update
When selected, overrides the compiler options found in the template being updated with the current compiler options.

Foreground processing always overrides the compiler options for older templates that do not contain compiler options.

You should select this option for batch template update of older templates to generate compiler options in the JCL.

Preserve copybook library
Ensures that, if a copybook still exists in the library that it was

previously found in and that library is in the list that the update is using, then that version of the copybook is used.

If you do not select this option, or the copybook no longer exists in the library it was previously found in, or that library is not in the list the update is using, then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

Show copybook name with record type field name
Select this option to make the copybook name for a record layout visible during template edit and for an edit, view or browse session where a copybook or template is being used.

Note: For the editor session where the layout name is specified it is prefixed with the name of the associated copybook. This may cause the layout name to be truncated, depending on its length.

Parent panels

- "Set Processing Options panel" on page 661
- "Set Language and Compiler Specifications (option 0.5)" on page 654

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Template types and structure" on page 127
- "Overriding Compiler Options" on page 145

Compare Utility: "Old" and "New" panels

The Compare Utility "Old" and Compare Utility "New" panels are used to specify the "Old" and "New" data sets in the comparison, and the conditions that form the *compare* subsets. Optionally, you can also specify the name of the copybooks or templates that describe logical views of the data sets and further refine the "Old" and "New" compare sets.

Both panels contain the same fields.

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Compare Utility                      Top of data
                                                                   More:
"Old" Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name  . . COPYBOOK                                          +
   Member  . . . . . . . . FMNCCPY       (Blank or pattern for member list)
   Volume serial . . . . . ______
   Start key . . . . . . . ________________________ key or slot
   Skip count  . . . . . . ________     number of records to be skipped
   Compare count . . . . . ALL          number of records to be compared

"Old" Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . . . . _________

Processing Options:
   Copybook/template usage           Enter "/" to select option
   3  1. Above                       _  Edit template   _ Type (1,2,S)
      2. Previous                    _  Advanced member selection
      3. None                        _  Skip member name list
      4. Create dynamic              _  Batch execution
                                     _  Binary mode, reclen _____

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward
 F8=Forward  F9=Swap     F10=Actions  F12=Cancel
```

Figure 134. Compare Utility - “Old” data set entry panel

“Old” (or “New”) Partitioned, Sequential or VSAM Data Set, or HFS file
You can use this panel to specify the data set, HFS file, WebSphere MQ queue name, or a CICS resource to be compared, and to define which records are to be compared. It contains:

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see “Specifying an MQ manager or queue” on page 21.

For information about specifying a CICS resource, see “Specifying a CICS resource” on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Note: The specification of a "New" member depends on the specification of an "Old" member. If the "Old" member specifies one member, the "New" member must also point to one member. If the "Old" member specifies a member name pattern, the "New" member must use the same pattern or an "*".

Volume
Specify a volume serial number if the data set is not cataloged.

Start key
VSAM only. Works in conjunction with **Skip count** to specify the first record from the data set to be included in the compare set.

If a start key or slot number is applicable to the data set, records are skipped until a record with a key or slot number greater than or equal to the specified value in the Start key field is found. If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. Keys can also be in hexadecimal format (for example, X'0102').

Skip count
Works in conjunction with **Start key** to specify the first record from the data set to be included in the compare set.

The number of records specified in the Skip count field is skipped after the start key or slot number (if applicable) has been honored.

Compare count
This field sets a limit on the number of records in the final compare sets that are included in the comparison.

“Old” (or “New”) Copybook or Template
These fields identify either a copybook or a template. If a copybook is specified, it is compiled to create a temporary template that can optionally be saved if edit was requested.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Processing Options
The processing options enable you to specify and modify a template that describes the logical view of your data set. The options are:

Copybook/template Usage

1. Above
Enter 1 to invoke data set compare with the specified copybook or template.

2. Previous
Enter 2 to invoke data set compare with the template last used with the specified input data set.

3. None
Enter 3 to invoke data set compare without a template.

4. Create dynamic
Enter 4 to create a dynamic template to be used when comparing.

Edit template
Specify / to edit the template before use. See Chapter 4, “Creating and editing templates,” on page 127 for detailed information about creating and editing templates.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

Skip member name list
Enter "/" to run without showing the member selection list.

Batch execution
Specify / to generate and edit a batch job to perform the comparison. For more information on editing these statements, see “DSM (Data Set Compare)” on page 934.

Note: This option is only available on the “Old” compare panel.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter “/” in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The field only displays if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. I/O exits can only be used to process the data sets that you are using. They cannot be used to process the copybook or template that you are using to format the data set.

Parent panels

The Compare Utility: “Old” panel is accessed from the “Utility Functions menu panel” on page 703. The Compare Utility: “New” panel can only be accessed from the “Old” panel.

Child panels

- “Compare Utility: Options panel” on page 479 (the “Old” and “New” data sets and templates, if specified, were fully qualified)
- “Data Set Selection panel” on page 519 (a pattern has been entered in the Data set name field)
- “Member Selection panel” on page 600(a pattern or a blank has been entered in the Member field)
- “Record Type Selection panel” on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- “Field Selection/Edit panel” on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).
- “Dynamic Template panel” on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- “Personal Data Set List panel” on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- “Personal Data Set Lists panel” on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

- “DSM (Data Set Compare)” on page 934

Related tasks and examples

- “Comparing data sets” on page 281

Compare Utility: Load module options panel

The Compare Utility - Load module options panel (Figure 135 on page 476) allows you to specify the way in which the load module comparison is processed and the type of listing produced.

Panel and field definitions

```
Process   Options   Help
------------------------------------------------------------------------------
 File Manager    Compare Utility : Load Module Options             Top of data

 Compare Options:
    Compare level      Listing type     Long Report
    1  1. Module       1  1. Summary    Enter "/" to exclude
       2. CSECT           2. Delta       _  Inserted
                          3. Matching    _  Deleted
 MQ Processing            4. Long        _  Changed
 _  Include descriptors   5. None        _  Matched
 Load module criteria:                  _ CSECT criteria:
    Enter "/" to select option             Enter "/" to select option
    _  Size                                _  Size
    _  Entry point address                 _  Address
    _  Linkage editor/Binder               _  Compiler
    _  Link date and time                  _  Compile date
    _  AMode/RMode                         _  AMode/RMode
    _  AC                                  _  IDR ZAP data
    _  Link attributes                     _  Text

 Processing Options:                     Listing Options:
    Enter "/" to select option              Enter "/" to select option
    _  Clear print data set                    Wide listing
                                            /  Highlight changes
                                            _  Show changed fields only
                                            _  YY/MM/DD (default: YYYY.DDD)

                                         Number of differences to report ALL

 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 135. Compare Utility : Load module options panel

Compare Options

Compare Level

1. Module
Only information on the load module level is extracted and compared. CSECT information (and differences on CSECT level) are ignored. This is a general comparison.

2. CSECT
Information on both levels, load module and CSECT, is extracted and compared. This is a more detailed comparison.

Listing type

1. Summary
Generates a summary listing only.

2. Delta
Generates a delta listing containing all deleted and inserted records, as well as a summary.

3. Matching
Generates a listing containing only matching records, and a summary. Matching records are only listed once.

4. Long
Generates a listing containing all matching and non-matching records. Matching records are only listed once.

5. None
Suppresses the listing. A message is issued to indicate whether or not the data sets match.

Long Report
Selecting one or more of these options causes the compare result types to be excluded from the report. These options are used in conjunction with the **Long** listing type to tailor the output report. If a listing type other than **Long** has been selected, these options are ignored.

Inserted
When selected, excludes inserted records from the report.

Deleted
When selected, excludes deleted records from the report.

Changed
When selected, excludes changed records from the report.

Matched
When selected, excludes matched records from the report.

MQ Processing

Include descriptors
If this is selected (with the / character), the message descriptor data returned by the GET MQ API is added as a prefix to the record data, allowing this data to be examined in conjunction with the message data.

Otherwise only the message data is available for processing.

Load module criteria
Selecting one or more of these options determines which load module properties are compared and reported. By deselecting some of the options, you exclude associated information from comparison. In this way, only differences of interest to you are shown; others are hidden.

Size Specify / to include the load module size in the comparison.

Entry point address
Specify / to include the load module entry point address in the comparison.

Linkage editor/Binder:
Specify / to include the version of the linkage editor or binder used to prepare the load module in the comparison.

Link date and time:
Specify / to include the load module link (bind) date and time in the comparison.

AMode/RMode:
Specify / to include the AMODE and RMODE of the load module in the comparison.

AC: Specify / to include the load module authorization code in the comparison.

Link attributes
Specify / to include the load module link attributes in the comparison.

CSECT criteria
Selecting one or more of these options determines which CSECT properties are compared and reported. By deselecting some of the options, you exclude associated information from comparison. In this way, only differences of interest to you are shown; others are hidden. These options are used in conjunction with the Compare level option. If CSECT has not been selected, these options are ignored.

Size Specify / to include the CSECT size in the comparison.

Address
Specify / to include the CSECT address in the comparison.

Compiler:
Specify / to include the version of the language compilers used to compile the CSECT in the comparison.

Date: Specify / to include the date of the CSECT compile in the comparison.

AMode/RMode:
Specify / to include the AMODE and RMODE of the CSECT in the comparison.

IDR ZAP data:
Specify / to include AMSPZAP IDR data in the comparison. Please note, that the IDR ZAP data constitutes separate records to be compared, but as it is associated with the CSECT, the IDR ZAP data should be seen as an extension of the CSECT data.

Text Specify / to include the CSECT content in the comparison. Please note, that the CSECT content is compared and then reported as a set of separate 32-byte records in "memory dump" format (hexadecimal and character), but as it is associated with the CSECT, the CSECT content should be seen as an extension of the CSECT data.

Processing Options

Clear print data set
Specify / to clear the File Manager print data set before use. This option has no effect if File Manager's print output is not directed to a data set. See “Setting your Print Processing Options” on page 299 for more information about controlling the print data set.

Listing Options

Wide listing
Specify / to generate a wide listing. Wide listing is easier to analyze as the corresponding data in the "New" and "Old" records are vertically aligned and the width of the record does not exceed the maximum record length supported by the printer.

Highlight changes
Specify / to highlight any differences between “Old” and “New” records. For record comparisons and wide formatted comparisons,

the differences are indicated with a change bar (|) beneath the altered bytes. For narrow formatted comparisons, the fields that have changed are indicated by placing an asterisk (*) to the left of the field name.

Show changed fields only
Specify / to report only changed fields (for a narrow-formatted report).

YY/MM/DD (default: YYYY.DDD)
Specify / to show dates in YY/MM/DD format. Otherwise, YYYY.DDD is used.

Number of differences to report
The number of differences after which the Compare Utility stops processing the data sets.

Parent panels

The Compare Utility: "New" panel.

Child panels

- None.

Equivalent functions

- "DSM (Data Set Compare)" on page 934

Related tasks and examples

- "Comparing data sets" on page 281

Compare Utility: Options panel

The Compare Utility - Options panel (Figure 135 on page 476) is used to specify the way in which the comparison is processed and the type of listing produced.

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
 File Manager              Compare Utility : Options

 Compare Options:
    Compare type      Synchronization    Listing type     Long Report
    1  1. Record      2  1. One-to-one   1  1. Summary    Enter "/" to exclude
       *. Formatted      2. Read-ahead      2. Delta        _  Inserted
                         3. Keyed           3. Matching     _  Deleted
                                            4. Long         _  Changed
                                            5. None         _  Matched

 Processing Options:                      Listing Options:
    Enter "/" to select option               Enter "/" to select option
       Edit template mapping                    _  Wide listing
    _  Clear print data set                     _  Show hex chars
    _  Create result data sets                  _  Highlight changes
    /  Ignore record length mismatch            _  Show field attributes
    _  Left justify numeric fields              _  Show changed fields only
                                                _  Always show SELECTed fields
    ISPF Packing
    1  1. Unpack if packed                   Number of differences to report ALL
       2. None
       3. Skip                               Template Reporting Options:
                                                Enter "/" to select option
                                                _  Show template layouts
                                                _  Show template criteria
                                                _  Show mapped fields
                                                _  Show unmapped fields
                                                _  Show array elements
                                                _  Show start/end as hex offset
                                                _  Show length in hex
                                                _  Show field statistics
 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 136. Compare Utility : Options panel

Compare Options

Compare Type

Choose either Record comparison or Formatted comparison.

1. Record

When the record comparison option is selected, pairs of records (determined by the record selection criteria and synchronization method) are compared byte by byte. For the records to match, they must be of equal length and the corresponding bytes in each record must be the same.

To use record identification and selection criteria to filter the records to be compared, specify a copybook or template for either the “Old” or the “New” data set or both.

Note: To perform a "record" comparison restricted to certain columns in the records, use dynamic templates to define the areas in the records to be compared using data type AN.

2. Formatted

When the formatted comparison option is selected, mapped pairs of fields (from each pair of records selected by the record selection criteria and synchronization method) are compared, according to the format defined for each field. For the records to match, every pair of mapped fields must match.

To specify field mapping, you must specify a copybook or template for both the “Old” and the “New” data sets. Optionally, you can also use the copybooks or templates to specify record identification and selection criteria to filter the records to be compared.

When this option is selected, pressing Enter displays the Formatted Comparison Options panel, in which you can specify additional comparison options. The display of this panel is "delayed" if read-ahead or keyed synchronization is requested, because the sub-option panels for synchronization are displayed first.

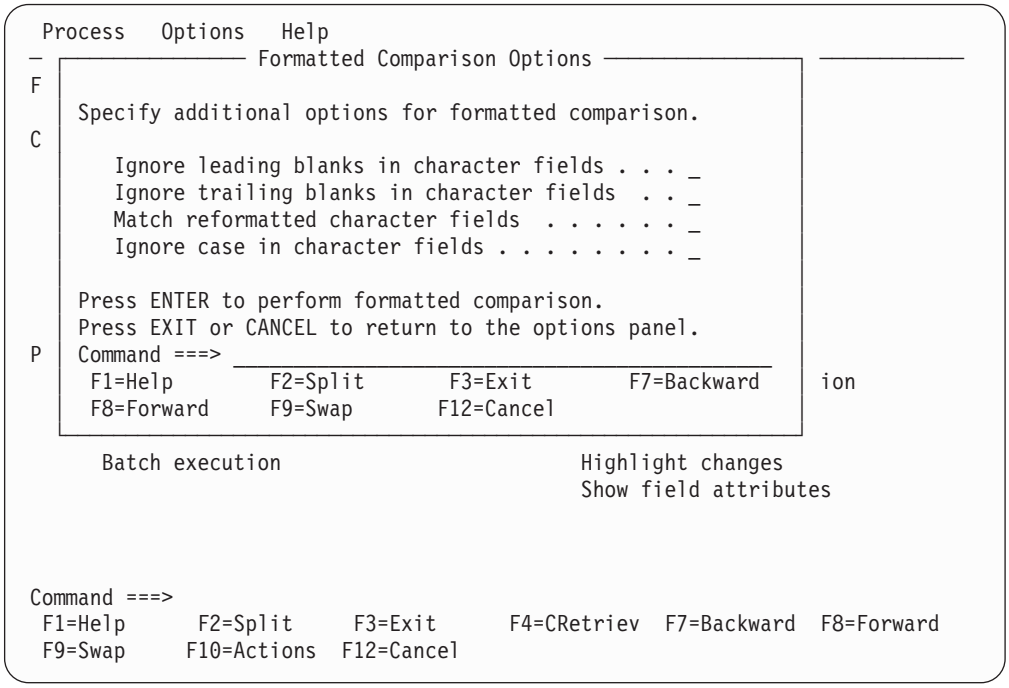


Figure 137. Formatted Comparison Options panel

Panel fields:

Ignore leading blanks in character fields
Select this option if you want leading blanks to be ignored when character comparisons are performed. This option is ignored if you also select the Match reformatted character fields option, because that option also causes leading blanks to be ignored.

Ignore trailing blanks in character fields
Select this option if you want trailing blanks to be ignored when character comparisons are

performed. This option is ignored if you also select the Match reformatted character fields option because that option also causes trailing blanks to be ignored.

Match reformatted character fields
Select this option if you want character fields to use a reformatted comparison method. Using this option, character fields match as if leading and trailing blanks were stripped, and all sequences of intermediate blanks were compressed to a single blank. The ignore leading and trailing blanks options are ignored if you select this option.

Ignore case in character fields
Select this option if you want character fields to match regardless of the case of the data being compared. Using this option character fields match as if all the alphabetic characters were converted to upper case.

Synchronization

1. One-to-one
The compare sets are assumed to contain records in a corresponding sequence.

2. Read-ahead
The compare sets are assumed to contain records in *roughly* corresponding sequences.

When this option is selected, pressing Enter displays an additional panel in which you can specify the Read-ahead synchronization limit and the Read-ahead synchronization length.

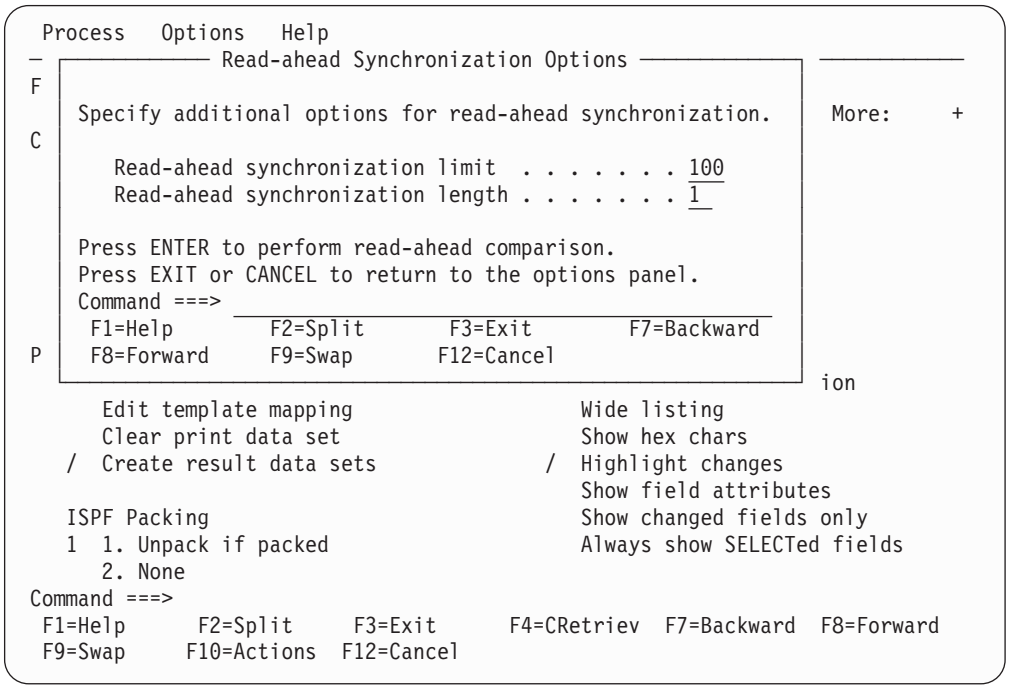


Figure 138. Read-ahead Synchronization Limit panel

Panel fields:

Read-ahead synchronization limit
Range: 1–999. Specifies the number of records to read ahead in each compare set.

Read-ahead synchronization length
Range: 1–99. Specifies the number of records that must match during read-ahead processing for synchronization to occur.

When non-matching records are found by the compare utility and read-ahead synchronization was specified, File Manager attempts to resynchronize the data by looking ahead in each of the data sets. The read-ahead length is the minimum number of matching records that must be found to recognize a resynchronization point. If insufficient matching records are found, read-ahead processing continues.

3. Keyed
The compare sets are assumed to be sequenced by a key, which is comprised of one or several segments of the record (up to sixteen).

When this option is selected, pressing Enter displays the Keyed Synchronization Settings panel, in which you must specify at least one key segment (and up to sixteen key segments). Any existing key segment information, such as that drawn from an intrinsically keyed data set or from key sequences specified in the “Old” or “New” templates, or both, is displayed in the panel. You can edit or modify these details.

```
 Process   Options   Help
─ ┌──────────────── Keyed Synchronization Settings ──────────────────┐ ──────────
F │                                                  Row 1 to 4 of 16 │
  │ Enter position(s), length(s) and type(s) for key segments.        │
C │                                                                   │
  │ Press ENTER to perform keyed comparison.                          │
  │ Press EXIT or CANCEL to return to the options panel.              │
  │                                                                   │
  │               Key position    Key     Key                         │
  │ Key segment   Old     New     length  type                        │
  │           1   _____   _____   _____   __                          │
  │           2   _____   _____   _____   __                          │
P │           3   _____   _____   _____   __                          │
  │           4   _____   _____   _____   __                          │ ion
  │                                                                   │
  │ Command ===> _____________________________________ Scroll CSR     │
  │  F1=Help       F2=Split       F3=Exit       F7=Backward           │
  │  F8=Forward    F9=Swap        F12=Cancel                          │ s
  └───────────────────────────────────────────────────────────────────┘


Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 139. Keyed Synchronization Settings panel - no key segment information

If you opt to use intrinsic keys from both your “Old” and “New” data sets and there is a length mismatch, File Manager uses the shorter of the two keys as the segment length. If there is any other type or length mismatch between key segments defined for the “Old” and “New” data sets, the Keyed Synchronization Settings panel contains additional columns that indicate the conflict. You can ignore the conflict and press Enter to process the panel using the values shown for the “Old” data set, or you can edit the “Old” key segment information and then continue.

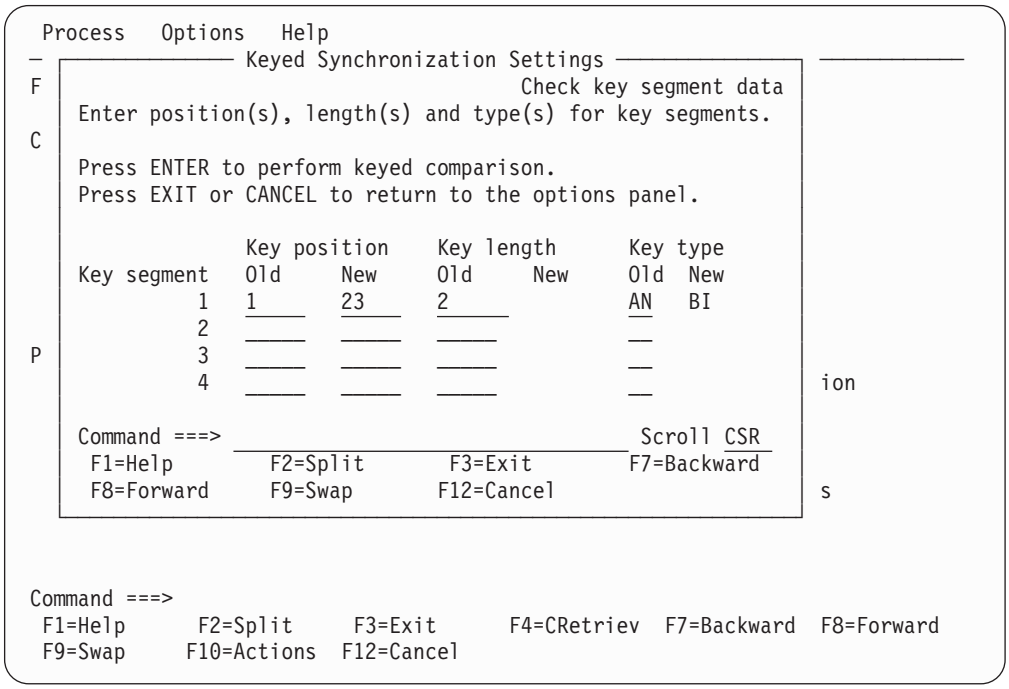


Figure 140. Keyed Synchronization Settings panel - conflicting key segment information

Panel fields:

Key position - Old
Range: 1-32760. Specifies the position of a key segment in the "Old" compare set records for keyed synchronization.

Key position - New
Range: 1-32760. Specifies the position of a key segment in the "New" compare set records for keyed synchronization.

Key length
Range: 1-32760. Specifies the length of a key segment for keyed synchronization.

Key type
Specifies the type of a key segment for keyed synchronization. The data type of the key segment must be one of the data types supported by the File Manager Compare Utility:

A/AN Alphanumeric

C/CH Character (same as alphanumeric)

B/BI Binary

P/PD Packed decimal

F/FP Internal floating point

Note: File Manager supports many other data types in other contexts.

If a data type is not explicitly specified, type AN is used.

Listing type

1. Summary
Generates a summary listing only.

2. Delta
Generates a delta listing containing all deleted and inserted records, and a summary.

3. Matching
Generates a listing containing only matching records, and a summary. Matching records are only listed once.

4. Long
Generates a listing containing all matching and non-matching records. Matching records are only listed once.

5. None
Suppresses the listing. A message is issued instead to indicate whether or not the data sets matched.

Long Report
Selecting one or more of these options causes those compare result types to be excluded from the report. These options are used in conjunction with the **Long** listing type to tailor the output report. If a listing type other than **Long** has been selected, these options are ignored.

Inserted
When selected, excludes inserted records from the report.

Deleted
When selected, excludes deleted records from the report.

Changed
When selected, excludes changed records from the report.

Matched
When selected, excludes matched records from the report.

Processing Options

Edit template mapping
Specify / to edit the template mapping before use. This option is only available if a copybook or template was specified for both the "Old" and "New" data sets, and is only valid if a formatted comparison is requested. See Chapter 4, "Creating and editing templates," on page 127 for detailed information about creating and editing templates and "Comparing data sets" on page 281 for details on using templates during the compare process.

Clear print data set
Specify / to clear the File Manager print data set before use. This option has no effect if File Manager's print output is not directed to a data set. See "Setting your Print Processing Options" on page 299 for more information about controlling the print data set.

Create result data sets
Specify / to create one or more output data sets. These data sets are specified on the Compare Utility: Output Data Sets panel.

Ignore record length mismatch
Enter "/" to ignore length mismatches with the template. If you do not select this option, records that are shorter or longer than the matching structure length in the template are not selected for processing.

Left justify numeric fields
If you specify "/" for this option and you have selected a Compare Type of **Formatted**, numeric fields are shown as left-justified.

Listing Options

Wide listing
Specify / to generate a wide listing. Wide listings are not suitable for printing if the width of the record or the width of the formatted data exceeds the maximum record length supported by the printer, but they can make it easier to analyze as the corresponding data in the “New” and “Old” records are vertically aligned.

Wide listings can accommodate records not exceeding 32K bytes in length. Longer records are truncated, but still marked correctly when changed.

Show hex chars
Specify / to generate a hexadecimal display of the records or formatted fields in addition to the standard character display.

Note: The Compare Utility and DCM function always use the UPDOWN format for hexadecimal display, regardless of the setting of the DUMP option in your Set Print Processing Options (option 0.1).

Highlight changes
Specify / to highlight any differences between “Old” and “New” records. For record comparisons and wide formatted comparisons, the differences are indicated by placing a "change bar" (|) beneath the altered bytes. For narrow formatted comparisons the fields that have changed are indicated by placing an asterisk (*) to the left of the field name.

Note: For altered numeric fields in a wide formatted comparison the change bars always indicate the entire field because the display characters do not necessarily correspond to individual bytes in the data.

Show field attributes
Specify / to show field attributes in formatted comparison reports. For a wide listing, leaving this option unchecked reduces the overall width of the report. For a narrow listing, leaving this option unchecked provides more space

in the formatted field area for the field name and data. This option has no effect for record type comparisons.

Show changed fields only
Specify / to report only changed fields if a narrow formatted report. This option has no effect for record type comparisons.

Always show SELECTed fields
Specify / to show SELECTed fields in formatted comparison reports. The option is intended to allow you to ensure that certain fields that you have SELECTed are always printed, regardless of whether they are changed or not. This option has effect only if the **Show changed fields only** option has been selected and only if the SELECTed fields have been mapped.

ISPF Packing
Provided that the data set is a sequential, PDS or PDSE file and an I/O exit routine is not used, one of these options can be used to control the comparison behavior when processing data that is in ISPF PACK format.

1. Unpack
If the data set is packed, it is unpacked before any processing or comparison is performed.

2. None
No checking or processing of ISPF packed data occurs. Packed records are processed and compared in packed format. This option is forced if an I/O exit has been used.

3. Skip
If the input data is packed, no processing or comparison occurs.

Template Reporting Options
These options control the template details reported by the Compare utility. If any of the template reporting options are specified, and template information participated in the comparison, a report of the relevant requested information appears after the comparison summary. If none of the template reporting options are specified, or no options relevant to the comparison are specified, no template information is reported.

Show template layouts
Specify / to show template field details in comparison reports. For each template used in the comparison, the template fields are listed with their attributes.

This option has no effect for record type comparisons, because template field information is not used in record type comparisons.

Show template criteria
Specify / to show template identification and selection criteria in comparison reports. For each template used in the comparison, the identification and selection criteria, if any, are listed.

Show mapped fields
Specify / to show how fields in the new data set template are mapped to fields in the old data set template.

This option has no effect for record type comparisons, because template mapping information is not used in record type comparisons.

Show unmapped fields
Specify / to show fields in the old and new data set templates that are not mapped to fields in the other template.

This option has no effect for record type comparisons, because template mapping information is not used in record type comparisons.

Show array elements
Specify / to show all occurrences of array elements in layouts. Array elements are not shown in mappings because mapping at the array element level is not supported.

For COBOL OCCURS DEPENDING ON tables, the maximum occurrences are shown.

For PL/1 REFER arrays, the minimum occurrences are shown.

Show start/end as hex offset
Specify / to show the start and end values as hexadecimal offsets in layouts and mappings.

Show length in hex
Specify / to show the length values as hexadecimal in layouts and mappings.

Show field statistics
Specify / to produce a field statistics report showing the number and percentage of mismatches by field.

Number of differences to report
The number of differences after which the Compare Utility stops processing the data sets.

Parent panels

- Compare Utility: “New” panel

Child panels

- “Compare Utility: Output Data Sets” on page 490

Equivalent functions

- “DSM (Data Set Compare)” on page 934

Related tasks and examples

- “Comparing data sets” on page 281

Compare Utility: Output Data Sets

You use the Compare Utility: Output Data Sets panel to specify the output data sets for containing records identified as inserted, deleted, changed, or matching that result from running the Compare Utility.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager       Compare Utility : Output Data Sets

Output Partitioned, Sequential or VSAM Data Set, or HFS file:

  Inserted Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

  Deleted Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

  New Changed Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

  Old Changed Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

  New Matched Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

  Old Matched Records
      Data set name  _____________________________________________  +
      Member . . . . ________
      Volume . . . . ______
      Disposition. . _       1. Old or reuse 2. Mod
      Binary mode. . _       Reclen _____

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 141. Compare Utility: Output Data Sets panel

Data set name
Specify a sequential, partitioned or VSAM data set, or HFS file. You can specify a generic data set name to select from a list.

Member name
Specify for a partitioned data set. Enter a blank or a pattern for a member selection list.

Volume
Specify a volume serial number if the data set is not cataloged.

Disposition:

Old or Reuse
Copies from the beginning of the existing data set.

Mod Appends the input records to the end of the data set. MOD is invalid for a member of a partitioned data set.

Binary mode
Data in an HFS file is processed without record delimiters (in binary mode). If the option is *not* selected, the data is assumed to be text and is processed as a collection of records delimited by EBCDIC New Line (X'15'), Carriage Return (X'25'), Line Feed (X'0D'), or their combinations.

Reclen
The logical record length used to deblock data into fixed records if **Binary mode** has been selected. The default record length is 80.

Parent panels

"Compare Utility: Options panel" on page 479

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Comparing data sets" on page 281

Copy From panel

You use the Copy From panel to specify the data set from which you want to copy records, the number of records to be copied, whether or not a template is to be used and various other processing options.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                    Copy Utility

 From Partitioned, Sequential or VSAM Data Set, or HFS file:
    Data set/path name  . . 'FMNUSER.EXPORT'                                  +
    Member  . . . . . . . . _________      (Blank or pattern for member list)
    Volume serial . . . . .           (If not cataloged)
    Start key . . . . . . . ____________________________ key or slot
    Skip count  . . . . . . _________ Include _________ Repeat skip ___________
    Copy count  . . . . . . ALL______ number of records to be copied
 From Copybook or Template:
    Data set name . . . . . 'FMNUSER.DATA'
    Member  . . . . . . . . TEMPA_____     (Blank or pattern for member)
 Processing Options:
    Copybook/template    Enter "/" to select option
    2  1. Above           _  Batch execution          _ Advanced member selection
       2. None            _  Use proc ________        _ Skip member name list
       3. Create dynamic  _  Ignore length mismatch   _ REXX member selection: P
    _  Edit template      _  JCL Source format        _ Directory integrity
    _  Type (1,2,S)       _  MQ Include descriptor    _ Report PDS record counts
    _  Export mode        _  MQ Get destructive       _ Binary mode, reclen _____
    _                     _  Use I/O exit
 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
  F9=Swap      F10=Actions  F12=Cancel
```

Figure 142. Copy Utility: "From" panel

From data set/file: Data set/path name
Specify the data set to be copied. Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

From Data Set: Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set to be copied. Required for data sets which are not cataloged.

Start key
Specifies the start position for the records to be copied (VSAM data sets only). Copying starts either at the first record with a key or slot number greater than or equal to the specified value in the Start key field, or after skipping the number of records specified in the Skip count field. If the key

contains lowercase characters, blanks, or commas, enclose it in quotation marks. Keys may also be in hexadecimal format (e.g. X'0102').

The Start key and Skip count fields are mutually exclusive.

Skip count
Number of logical records to be skipped from the beginning of the file. By default processing begins with the first record in the file.

Range: 0-99999999. Default: 0.

Include
Represents the number of physical records that to be included in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Range: 0-9999999. Default: 0.

Repeat skip
Represents the number of physical records to be skipped in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Range: 0-9999999. Default: 0.

Copy count
Specify the number of logical records to be copied.

Range: 1-99999999, ALL. Default: ALL.

From Copybook or Template: Data set name
Data set name of the template or copybook to be used to select and format the data to be copied. This field is ignored when the Copybook/template usage field is set to **2. None**.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

From Copybook or Template: Member
If you specified the name of a partitioned data set (PDS) or CA-Panvalet or other external library in the **Data set name** field, then use this field to specify the member name or member name pattern. This field is ignored when the Copybook/template usage field is set to **2. None**.

Copybook/template usage
Indicates if you want to use a template to format and select your data.

1. Above
Use the template specified on the panel (or compile the specified copybook into a template, and use that).

2. None
No logical view is to be used by the function.

3. Create dynamic
Create a dynamic template.

Edit template
Indicates that you want to edit the template before use. You edit the template when you need to change format, selection, data create attributes, and reformatting information for output copy templates.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Export mode
Indicates that the input dataset (file) is to be copied into the output dataset (file) in an external format.

Batch Execution
Allows you to edit the JCL to run the function in batch.

Use proc
Allows you to specify a procedure (containing REXX or DFSORT statements or both) to run in conjunction with the Copy action.

This option has two fields. To select the option, enter "/" in the field to the left of **Use proc**. With this selected, you can then specify which procedure to use in the field to the right of the field label (the proc name field). In the name field you can perform either of these actions:

- Enter a temporary procedure for one-time use, by entering a single asterisk (*). File Manager displays an Edit panel that you can use to create a new procedure.
- Specify the name of the member containing the procedure you want to use. The member must belong to the PDS allocated to ddname FMNEXEC. You can enter any of the following:
  - The name of the member.
  - A member name pattern (other than a single *) to list all matching members. You can then select the required member by entering an S in the **Sel** field. A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

    asterisk (*)
    Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter `*d*`, a list of all members in the data set whose name contains "d" is displayed.

    percent sign (%)
    A place-holding character representing a single character. As many percent symbols as necessary can appear anywhere in a member name. For example, if you enter `%%%`, a list of all members in the data set whose name is four characters in length is displayed.

Ignore length mismatch
Indicates that you want to ignore length mismatches with the template. If you do not specify this, records that are shorter than the matching structure length in the template are not selected for processing. If you select **Ignore length mismatches**, then "from" fields that span the record boundary in the input data set record have their corresponding fields on

the output record initialized, with the exception of alphanumeric fields where a partial copy is done for the portion of the field that exists on the input record.

JCL Source format
Indicates the file or PDS(E) members being copied contain JCL, the syntax of which is to be maintained during the copy process. You can only use this option if you have selected the **Use proc** option.

Note: Use of this option may affect the copy performance due to the processing overhead involved in verifying the syntax of the JCL.

MQ Include descriptor
If this is selected (with the / character), the message descriptor data returned by the GET MQ API is added as a prefix to the record data, allowing this data to be examined in conjunction with the message data.

Otherwise only the message data is available for processing.

MQ Get destructive

If this is selected (with the / character), the input messages queue (if an MQ queue is specified) is opened in SHARE mode and the MQ API GET destructively reads each message.

Otherwise the queue is opened in BROWSE mode and messages are not removed from the queue. This is only effective for the DSC (Copy) function.

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of **Use I/O exit**. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The field only displays if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of **Use I/O exit**.
2. I/O exits can only be used to process the data sets that you are using. They cannot be used to process the copybooks or templates that you are using to format the data sets.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

Skip select member panel
Enter "/" to run without showing the member selection list. This option is ignored if errors are found whilst copying (for example, duplicate member names when renaming members).

REXX member selection
Enter "/" to use a REXX procedure to determine if a member should be copied.

If this option is selected, then the **Use Proc:** option must be selected, a REXX procedure must be supplied and you can (optionally) specify the default action to be applied in the event the nominated procedure fails to determine if a member in the input data set should be copied.

The default action is determined by entering a `P` (PROCESS MEMBER) or `D` (DROP MEMBER) in the adjacent field. If no default is provided, P is assumed.

Directory integrity
Forces an override of the default PDS(E) member processing method which allows for faster PDS directory access.

This option has significant performance impact. When selected, the members are processed in a way which allows concurrent directory updates as File Manager accesses the members using current directory information.

When not selected, the member processing is performed faster, but may be affected by PDS(E) directory updates, possibly causing I/O errors if the data set is updated concurrently.

Report PDS record counts
The count of records for copied PDS(E) members and sequential/VSAM data sets is printed in the processing listing in batch. Copy progress counts are displayed on screen while processing in online mode.

Note: This option affects PDS(E) processing ONLY. For sequential/VSAM data sets, the record counts are always provided. When the option is selected, it prevents the use of IEBCOPY for PDS(E) processing, which may affect the copy performance.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Parent panels

"Utility Functions menu panel" on page 703

Child panels

- "Copy To panel" on page 497 (the Copy From data set and template, if specified, was fully qualified)
- "Data Set Selection panel" on page 519 (a pattern has been entered in the Data set name field)
- "Member Selection panel" on page 600(a pattern or a blank has been entered in the Member field)
- "Record Type Selection panel" on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- "Field Selection/Edit panel" on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).

- "Dynamic Template panel" on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

"DSC (Data Set Copy)" on page 875

Related tasks and examples

- "Copying data sets" on page 253
- "Supplying a procedure when using a File Manager panel" on page 417
- "Selecting a range of PDS(E) members" on page 34

Copy To panel

You use the Copy To panel to specify the data set to which you want your copy to be made. This panel comes in three versions, depending on whether or not you have specified a template in the Copy From panel and requested that the output be in an external format.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
Copy from USERID.FMDATA(DATA1)

To Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set name . . . . . FMDATA                                          +
   Member  . . . . . . . . DATA2
   Volume serial . . . . . ______

Processing Options:
   Disposition             Execution "/" options          ISPF Packing
   1  1. Old or Reuse       _  Replace members            1  1. Asis
      2. Mod                _  Binary mode, reclen _____     2. Pack
   Stats Option                                             3. Unpack
   1  1. Off                                                4. None
      2. Force                                              5. Skip









Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 143. Copy Utility: "To" panel (no copybook or template in Copy From panel)

Copy from
Lists the data set, HFS file (directory), WebSphere MQ queue name, or a CICS resource from which you are copying data.

To data set/file: Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

To Data Set: Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which is to contain the copied data set. Required for data sets which are not cataloged.

Disposition
The To data set status:

Old/Reuse
Writes copied records into the output data set, starting from the beginning of the set and replacing any existing records.

Mod Appends the input records to the end of the data set unless the data set is a VSAM KSDS data set. In this case, the records are positioned in the data set using the key value in the input record. MOD is invalid for a member of a partitioned data set.

Replace members
Replace like-named members in an output partitioned data set.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Stats Determines whether ISPF statistics (if present) for the PDS members being processed are updated:
Blank Update ISPF statistics.
1 Off Do not update ISPF statistics.
2 Force
Always update or create ISPF statistics.

Use I/O exit

Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The field only displays if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. I/O exits can only be used to process the data sets that you are using. They cannot be used to process the copybook or template that you are using to format the data set.

ISPF Packing

Provided that the output data set is a sequential, PDS or PDSE file, an I/O exit routine is not used and the DISP is set to OLD, one of these options can be used to control the copy behavior when processing data that is in ISPF PACK format.

1. Asis

If the input data set is packed, it is unpacked before any processing. The output is written in packed format only when the input is packed.

2. Unpack

If the input data set is packed, it is unpacked before processing. The output is always written in unpacked format.

3. Pack

If the input data set is packed, it is unpacked before processing. The output is always written in packed format.

4. None

No checking or processing of ISPF packed data occurs. This option is forced if an I/O exit has been used.

5. Skip

If the input data is packed, no processing or copying occurs.

When you have specified a template in the Copy From panel, the Copy To panel contains additional fields, so that you can choose to specify a template for the output data set.

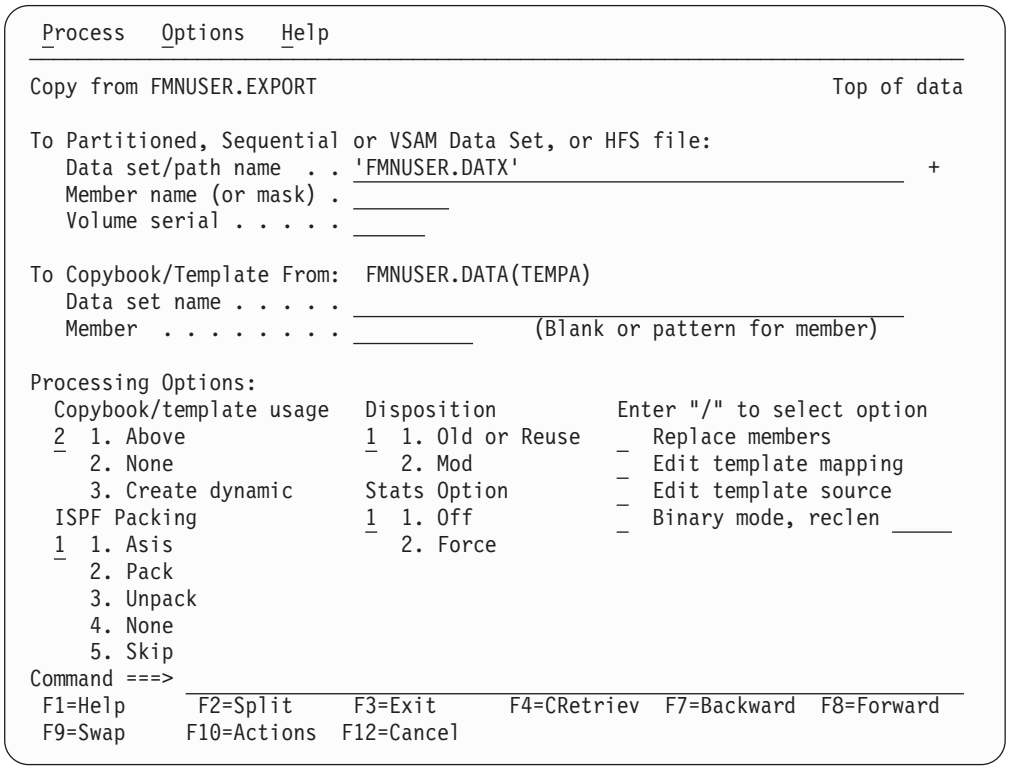


Figure 144. Copy Utility: "To" panel (copybook or template specified in Copy From panel)

Copybook/Template From
Lists the copybook or template specified in the Copy From panel, if applicable.

To Copybook or Template: Data set name
Data set name of the template or copybook to be used when mapping fields from the From Copybook or Template. This template does not affect the record selection or field format of the copied data. The field is ignored when the Copybook/template usage field is set to **2. None**.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

To Copybook or Template: Member
If you specified the name of a partitioned data set (PDS) or CA-Panvalet or other external library in the **Data set name** field, then use this field to specify the member name or member name pattern. This field is ignored when the Copybook/template usage field is set to **2. None**.

Edit template mapping
Specifies that you want to change the mapping of input fields to output fields or the data creation patterns for new fields.

Edit template source
Specifies that you want to edit the copybook used to generate the template.

If you selected the **Export mode** option on the Copy From panel, the Copy To panel contains additional fields specific for external format that allow you to customize the result of copy.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 Copy from FMNUSER.FMDATA

 To Partitioned, Sequential or VSAM Data Set, or HFS file:
    Data set/path name  . . 'FMN.FMDATA'                                  +
    Member name (or mask) . ________
    Volume serial . . . . . ______

 Processing Options:
   Disposition           Execution "/" options        Non-print. characters
   1  1. Old or Reuse     _  Replace members           2  1. Asis
      2. Mod              _  Binary mode, reclen          2. Hex
   ISPF Packing           _  Include fillers              3. Replace with .____
   1  1. Asis             _  Include redefines            4. Skip
      2. Pack             _  Convert to Unicode        Special characters
      3. Unpack           _  Split output line         1  1. Escape
      4. None                                             2. CData
      5. Skip                Stats Option                 3. Hex
   Format                    1  1. Off                    4. Replace with *____
   1  1. XML                    2. Force                Invalid data
                             Indent step 1                1  1. Hex
                                                             2. Replace with *____
                                                             3. Skip
 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 145. Copy Utility: "To" panel (Export mode specified in Copy From panel)

Disposition

1. Old or Reuse
Copies from the beginning of the existing data set.

2. Mod
Appends the input records to the end of the data set. MOD is invalid for a member of a partitioned data set.

ISPF Packing
Provided an I/O exit routine is not used, one of these options can be selected to control the copy behaviour when considering data that is in ISPF packed format. (Note that packed options do not apply to VSAM input.)

1. Asis
If the input data set is packed, it is unpacked before any processing. The output is written in packed format if the input was packed.

2. Pack
If the input data set is packed, it is unpacked before processing. The output is written in packed format.

3. Unpack
If the input data set is packed, it is unpacked before processing. The output is written in unpacked format.

4. None
No checking or processing of ISPF packed data occurs. This option is forced if an I/O exit has been used.

5. Skip
If the input data is packed, no processing or copying occurs.

Format
Specifies an external format to be used for the output.

1. XML
The output is generated in XML format.

Execution "/" options

Replace members
Replace like-named members in an output partitioned data set.

Binary mode
Data in an HFS file is processed without record delimiters (in binary mode).

reclen: The logical record length used to deblock data into fixed records (default: 80).

Include fillers
Indicates whether fillers (unnamed items), defined in the input template (COBOL copybook or PL/I include), are to be represented in the output or ignored.

Include redefines
Indicates whether redefinitions of data items, specified in the input template (COBOL copybook or PL/I include), are to be represented in the output or ignored.

Convert to Unicode
Indicates whether the output is to be converted to Unicode, or not.

Split output line
Indicates whether the output lines resulting from processing an input record are to be spanned contiguously over multiple output records. If so, the output records will not match output lines. If not, each output line must fit (as the only line) into a single output record, otherwise File Manager truncates the output, ends processing, and reports an error.

Use I/O exit
Activates the specified exit routine which handles a compressed or encrypted data set. Specify the name of the exit routine unless the default exit is in effect and can be used.

Indent step
Defines the number of blanks used to indent each level of XML tag nesting (each nested level in the template, COBOL copybook, or PL/I include causes an increase in indentation by the specified number of blanks). Valid range: 0–9 (default:1).

Non-print. characters
Indicates how non-printable characters are to be represented in the output.

1. Asis
Non-printable characters appear unchanged in the output.

2. Hex A value with one or more non-printable characters is substituted by its hexadecimal representation.

3. Replace with *replacing-character*
Each non-printable character is substituted with a *replacing character*, or each substring of non-printable characters is converted to its hexadecimal representation and surrounded by nested <HEX> and </HEX> tags. The set of allowable replacing characters is limited to printable characters with the exception of special characters.

You can specify the replacing character in one of the following forms:

char Each non-printable character is replaced with a character, such as "?". Default character: "." (dot).

C'*char* Each non-printable character is replaced with a character without case translation.

X'*cc***'** Each non-printable character is replaced with a character defined by its hexadecimal value.

HEX If you specify HEX instead of a replacing character, each substring of consecutive special characters is replaced by its hexadecimal representation, tagged by <HEX> and </HEX>, and nested into the content of the element. In other words, each string of consecutive special characters is represented by:

```
<HEX>hex-representation-of-string-of-non-printable-characters</HEX>
```

nested in the content of the element.

4. Skip
Value is skipped if it contains any non-printable characters.

Special characters
Indicates how special characters are to be represented in XML output.

1. Escape
Special characters are converted into escaped strings:
">" for ">"
"<" for "<"
"'" for "'"
""" for """
"&" for "&"

2. CData
The string containing special characters is left unchanged. It is enclosed in the CDATA section.

3. Hex A value with one or more special characters is substituted with its hexadecimal representation.

4. Replace with *replacing-character*
Each special character is substituted with a *replacing character*, or each substring of special characters is converted to its hexadecimal representation and surrounded by nested <HEX > and </HEX > tags. If a replacing character is specified or defaulted, each special character is substituted with the replacing character. The set of allowable replacing characters is limited to printable characters with the exception of special characters.

You can specify the replacing character in one of the following forms:

char Each special character is replaced with a character, such as "?". Default character: "_" (underscore).

C'*char***'** Each special character is replaced with a character without case translation.

X'*cc***'** Each special character is replaced with a character defined by its hexadecimal value.

HEX If you specify HEX instead of a replacing character, each substring of consecutive special characters is replaced by its hexadecimal representation, tagged by <HEX> and </HEX>, and nested into the content of the element. In other words, each string of consecutive special characters is represented by:

```
<HEX>hex-representation-of-string-of-special-characters</HEX>
```

nested in the content of the element.

Invalid data

Indicates how invalid data is to be represented in the output.

1. **Hex** Any invalid value is substituted by its hexadecimal representation.
2. **Replace with** *replacing character*
   Any invalid value is replaced with a string of *replacing characters* for the length of the value. The set of allowable characters is limited to printable characters with the exception of special characters.

   You can use:

   char The value is replaced with a string of characters, such as "?". Default character: "*" (asterisk).

   C'*char***'** The value is replaced with a string of characters without case translation.

   X'*cc***'** The value is replaced with a string of characters, each character being defined by its hexadecimal value.
3. **Skip**
   Any invalid value is skipped.

Parent panels

- "Copy From panel" on page 491

Child panels

- "Copy From panel" on page 491 (the Copy To data set and template, if specified, was fully qualified)
- "Data Set Selection panel" on page 519 (a pattern has been entered in the Data set name field)
- "Member Selection panel" on page 600(a pattern or a blank has been entered in the Member field)
- "Record Type Selection panel" on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- "Field Selection/Edit panel" on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).
- "Dynamic Template panel" on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

"DSC (Data Set Copy)" on page 875

Related tasks and examples

- "Copying data sets" on page 253
- "Supplying a procedure when using a File Manager panel" on page 417

Copybook Selection panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Copybook Selection

Cmd R Member    Lib 01 Field Name
*** ****  Top of data  ****
___   MID1        1 _  ____________________________________________________________
___   COPYCONC    1 _  ____________________________________________________________
___   COPYMM      1 _  ____________________________________________________________
___   FLMLDATE    1 _  ____________________________________________________________
___   FMNCPY1     1 _  ____________________________________________________________
*** ****  End of data  ****









Command ===> ____________________________________________________ Scroll HALF
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=LibList   F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 146. Copybook Selection panel

Cmd Prefix command field. Allows you to edit, view, copy, insert or delete members. Available commands are:

- **E** Create (if necessary) the template and edit it.
- **A** Identifies the record after which data is to be moved or copied.
- **B** Identifies the record before which data is to be moved or copied.
- **C** Copy one record.
- **C***n* Copy *n* records.
- **CC** Copy block of records. Marks the start and end of the block.
- **D** Delete one record.

D*n* Delete *n* records.

DD Delete block of records. Marks the start and end of the block.

I Insert one empty record.

I*n* Insert *n* empty records.

M Move one record.

M*n* Move *n* records.

MM Move block of records. Marks the start and end of the block.

R Repeat record once.

R*n* Repeat record n times.

RR Repeat block of records. Marks the start and end of the block.

RR*n* Repeat block of records *n* times. Marks the start and end of the block.

S Edit the member. (Restricted to members stored in a PDS or PDSE.)

U Update and edit the template.

V View the member.

X Displays the Redefines / Range Specifications panel.

R This column contains the value "R" if redefines or range parameters have been specified for the corresponding copybook member.

Member
Copybook member name field. If a mask is provided as a member name, the Member Selection panel is displayed, allowing one or more members to be selected for inclusion in the process list at the current position.

Note: All members must use the same language, either COBOL or PL/1. If you specify a member that consists of an entire program, from which File Manager is to extract the copybook information, you should only specify one member. Mixing a program member with other copybook members might cause compilation errors.

Lib The sequence number of the library data set containing the member, as specified on the Library List panel.

01 Flag field (Y) to specify that File Manager should insert a 01-level definition into the template at the beginning of the copybook details provided in the member.

Field name
01-level name of up to 35 bytes to be inserted when the 01 field contains a "Y". If a member is flagged as requiring a 01-level definition and no 01-level name field is supplied, one is generated.

You can use the following primary commands on this panel:

- CANCEL
- END
- LIBLIST
- LOCATE
- UPDATE
- EDIT

Parent panels

- "Library List panel" on page 588

Child panels

- "Member Selection panel" on page 600
- "Template Save pop-up panel" on page 697
- "Redefines / Range Specifications panel" on page 638

Related tasks and examples

- "Advanced copybook selection" on page 147

Copybook View and Print: Entry panel

The Copybook View and Print: Entry panel is used to specify the template definition that you want to view or print. These field definitions can be stored in:

- a copybook
- a template that was based on a copybook (not DB2 templates)
- a dynamic template
- an IMS view
- an IMS criteria set

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager           Copybook View and Print

Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . . . . __________   Blank or pattern for member list

Processing Options:
   View/Print Option                 Enter "/" to select option
   1  1. View                          Batch Execution
      2. Print                       /  Show array elements
                                     _  Show start/end as hex offset
                                     _  Show length in hex
                                     _  Show template criteria









Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 147. Copybook View and Print Utility panel

Copybook or Template

The name of copybook, template (non DB2), IMS view or IMS criteria set that you want to view or print, which is specified using:

Data set name

The name of the sequential or partitioned data set that contains the copybooks, templates, IMS views or IMS criteria sets that you want

to view or print. Can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Note: A copybook must be a member of a partitioned data set. Templates, IMS views and IMS criteria sets can be in a sequential data set.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Processing options
You can specify whether to view or print the layout and also request batch or foreground execution. The options are described below:

View/Print Option
This option is ignored for batch execution.

1. View
Displays the report on a scrollable panel.

2. Print
Prints the report to the current print data set.

Batch execution
Runs the function in batch. The **Set Batch Job card Information** panel can be used to tailor the default JOB card that File Manager uses to generate JCL.

Show array elements
Shows all occurrences of array elements.

For COBOL OCCURS DEPENDING ON arrays, the maximum occurrences are shown. For PL/1 REFER arrays, the minimum occurrences are shown.

Show start/end as hex offset
Shows start and end values as hexadecimal offsets.

Show length in hex
Shows length values as hexadecimal.

Show template criteria
Shows any identification and selection criteria contained in the template.

If a copybook is being viewed or printed, this option has no effect.

Parent panels

- "Utility Functions menu panel" on page 703
- "Template and Copybook Utility functions panel" on page 687

Child panels

- "Copybook View and Print: View panel" on page 509

Equivalent functions

- None.

Related tasks and examples

- "Copybook View and Print Utility (option 3.13 or 7.2)" on page 193

Copybook View and Print: View panel

The Copybook View and Print: View panel displays the field information stored in a selected template or copybook.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager      USERID.COPYBOOK(COPY0102)                     Line 1 of 18

1 Print Layout

Ref Field Name                                  Picture  Type Start    End Length
    ****  Top of data  ****
  1 1 REC-TYPE01                                         AN      1     80     80
  2  2 REC-TYPE                                 XX       AN      1      2      2
  3  2 NAME                                     X(20)    AN      3     22     20
  4  2 EMPLOYEE-NO                              9(4)     BI     23     24      2
  5  2 AGE                                      9(4)     BI     25     26      2
  6  2 SALARY                                   9(7)     PD     27     30      4
  7  2 MONTH OCCURS 12 TIMES                    9(8)     BI     31     34      4
  8  2 FILLER                                   XX       AN     79     80      2

  1 1 REC-TYPE02                                         AN      1     80     80
  2  2 REC-TYPE                                 XX       AN      1      2      2
  3  2 NAME                                     X(20)    AN      3     22     20
  4  2 JOB-TITLE                                X(14)    AN     23     36     14
Command ===>                                                    Scroll 0001
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 148. Copybook View and Print: View panel

This panel displays the layout for the copybook or template you referenced via the entry panel. You can print the contents of the display to the current print data set.

Note: The panel shown in Figure 148 shows a template derived dynamically from a copybook, in which case, criteria expressions cannot be present.

However, if the **Show template criteria** option on the Copybook View and Print entry panel has been selected, for a precompiled template criteria expressions may be shown on this panel.

Field reference
: Shows the field reference number assigned by File Manager to the field name. Field reference numbers are assigned to all fields in a record, and are used to identify fields in identification and selection criteria expressions and in primary commands. For IMS templates, views and criteria sets, the reference is suffixed with a K to flag key fields.

Field name
: Shows the level number and field name of the field. Various other attributes, such as redefines, array sizes, and depending on references, are also displayed as part of the name. Non-zero offset values are displayed

for Level 1 fields. For IMS templates, views and criteria sets, the Level 1 field name includes the segment reference.

Picture
Shows the picture clause for COBOL and for PL/I shows the length and scale (if non zero) for binary and packed fields. Also shows the bit length for bit fields.

Type Shows the data type of the field:

AN Alphanumeric (includes alphabetic, group items, edited PL/I numeric picture fields, and edited COBOL numeric picture fields where not all elements of the picture clause are supported).

AX Alphanumeric displayed in long hexadecimal.

BI Binary

BT Bit

FE Floating point (external)

FP Floating point (internal)

G/DB Graphic/DBCS

PD Packed decimal (internal decimal)

VB Varying bit

VC Varying character

VG/VD
Varying graphic/DBCS

ZC Z Character

ZA Zoned Alphanumeric (COBOL external edited). COBOL edited picture definitions that do not fit the ZE definition.

ZD Zoned Decimal (external decimal)

ZE Zoned Edited (COBOL external edited). Supports COBOL picture edit characters `+ - . 9`

ZG/Z2 Varying graphic/DBCS null terminated

Start Shows the start column of the field. For variable located fields, the start location is based on the maximum length of the record.

End Shows the end column of the field.

Length
Shows the length of the field. For a Level 01 field, if the record is variable length, the length shown is the maximum record length.

Available commands

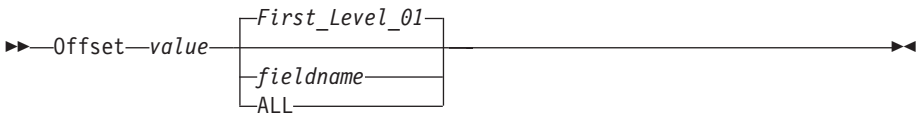
1 or P Prints the displayed report to the print data set.

FIND Finds the next occurrence of a character string in the data being displayed, starting at the top of the displayed page or starting at the cursor position (if the cursor is within the data portion of the display). The cursor is placed at the beginning of the found string. Automatic scrolling is performed, if necessary, to bring the string into view. See “FIND/FX primary command” on page 769 for more details about this command and its syntax.

Offset Repositions the field start locations by the value specified. The length of the Level 01 field is adjusted by the offset and the remaining fields within the layout have the offset added to their start location.

Note: Using the OFFSET commands for IMS data set members invalidates key field references, as these cannot be validated in the base product. Key field references remain as they were derived in the originating template, view or criteria set.

Syntax



value The length of Level 01 fields in the report and subsequent fields within the layout is adjusted by the value provided, which must be in the range -32760 to 32760. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the report contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the report.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the report.

RESet This command is used to reset any field highlighted by the FIND command.

Parent panels

- "Copybook View and Print: Entry panel" on page 507

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Copybook View and Print Utility (option 3.13 or 7.2)" on page 193

Create panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                        Create

To Data set:
   Data set/path name  . . 'FMN.FMDATA'
   Member name (or mask) . ________
   Volume serial . . . . . ______

Allocation Options:
1  1. Allocate using the attributes of
        'FMNUSER.DATA'
   2. Specify allocation attributes

Processing Options:
ISPF Packing                         Enter anything to select option
1  1. None                           _  Binary mode, reclen _____
   2. Pack



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 149. Create panel

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

The **Allocation Options** are only required when the "To" data set does not exist.

The **Binary mode** and **reclen** parameters are only relevant for an HFS file.

Parent panels

Child panels

- "Allocate panel" on page 448
- "Allocate (Two) panel" on page 451
- "Allocate (Three) panel" on page 453

Equivalent functions

- None.

Related tasks and examples

- "Creating, replacing, and adding data using existing data" on page 246

Create Tape Data panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Create Tape Data

Tapes:  No tapes allocated


Input:
 DDNAME to use . ________              enter new name, or select one from above
 Tape mode . . . __                    optional recording mode or density code

Output:
 Records . . . . ________              number of records
 Record length . 50
 Fillchar  . . . _____                 char or hex value, AN, BIN, or RAND
 Sequence field position  . _______    if sequence field desired
 Sequence field length  . . 8          length from 1 to 9
 Sequence field increment . 10         increment value
 Record format . ___                   F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
 Block size  . . _______               required for blocked output

Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 150. Create Tape Data panel

Tapes List of tapes currently allocated.

DDNAME to use
Specify the DDNAME used to allocate a tape unit. File Manager displays the DDNAME and VOLSER of an allocated tape unit, if any.

If the DDNAME specified is not allocated, File Manager prompts for allocation information.

Records
Number of logical records to be written. Range: 0-99999999

Record length
Length of the records to be written. Range: 1-9999999, depending on the output file characteristics.

Fill char
To fill each byte of each record with data, specify one of the following:

char A character such as 0 to be written in each byte.

X'cc' A binary character, such as X'04' to be written in each byte.

AN To write alphanumeric characters (A to Z, 0 to 9).

AX Alphanumeric displayed in long hexadecimal.

BIN To write binary characters (X'00' to x'FF').

RAND
To write random binary characters (X'00' to x'FF').

Default: blank character

Sequence field position
Position (starting with 1) of the sequence field within the output block or record. If omitted, there is no sequence field. Key position + key length - 1 must be less than or equal to the record length.

Range: 1 to *n* where

n <= record length - key length + 1

Default: No key

Sequence field length
Length of the sequence field. Key position + key length - 1 must be less than or equal to the record length.

Range: 1-9

Default: 8

Sequence field increment
Increment value used to increase the number in the sequence field.

Range: Any positive numeric value fitting into the sequence field.

Default: 10

Specify one of the following:

U undefined

F fixed

FB fixed blocked

V variable

VB variable blocked

VS variable spanned

VBS variable blocked spanned

D ASCII variable length

DB ASCII variable blocked

DS ASCII variable spanned

DBS ASCII variable blocked spanned

The actual block size for record format F and FB, otherwise the maximum block size. For record format FB or VB a block size is required, otherwise it is optional.

Range: 1-9999999 for F,FB,U records, 9-65535 for V,VB records

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

- None.

Equivalent functions

- "BT (Create Tape File)" on page 838

Related tasks and examples

- "Create Tape Data (option 4.7)" on page 360

Data Create Utility panel

You can use the Data Create Utility panel to create and initialize sequential data sets, VSAM data sets, and PDS data sets and members.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                Data Create Utility                  Top of data

Output Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name  . . . /u/testonly/xxx.txt                              +
   Member  . . . . . . . . . *          (Blank or pattern for member list)
   Volume serial . . . . . . ______     (If not cataloged)
   Record length . . . . . . 90         Optional record length for RECFM V
   Records . . . . . . . . . ________   Number of records
   Fillchar  . . . . . . . . ______     Char or hex value, AN, BIN, or RAND
   Sequence field position . ______     If sequence field desired
   Sequence field length . . 8          Length from 1 to 9
   Sequence field increment  10         Increment value
   Like data set . . . . . . ____________________________________________
Copybook or Template:
   Data set name . . . . . . 'FMNUSER.TEMPLATE'
   Member  . . . . . . . . . _________  (Blank or pattern for member list)
Processing Options:
   Disposition       Copybook or template     Enter "/" to select option
   2  1. Old/Reuse   1  1. Above              _  Edit template   _ Type (1,2,S)
      2. Mod            2. Previous           _  Batch execution
   ISPF Packing         3. None               _  Binary mode, reclen _____
   1  1. None           4. Create dynamic
      2. Pack
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 151. Data Create Utility panel

Data set/path name
Can be a fully-qualified data set name or a pattern, or a HFS file or directory. The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Note: This can be the name of a new or existing data set. If it is a new data set, you are asked to allocate the data set when you process the panel.

Member
If you specified the name of a partitioned data set (PDS) without including

a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which is to contain the copied data set. Required for data sets which are not cataloged.

Record length
The length of the records to be written. For variable record formats, the value specified is used when it is less than the maximum record size. Otherwise, the maximum record size is used and user input is ignored. For fixed record formats, the value is set to the data set record length and user input is ignored. For undefined record formats, the value is set to the block length and user input is ignored.

Records
The number of records to be written to the output data set.

Fillchar
Determines how each byte of each record is to be filled. You can enter:

char Writes the character *char* in each byte. Alphabetic characters are translated to upper case.

C'*char***'** Writes the character char in each byte without case translation.

X'cc' Writes X'cc' in each byte.

AN Fills the record with consecutive characters (A to Z, 0 to 9)

BIN Fills the record with consecutive binary characters (X'00' to X'FF')

RAND
Fills the record with random binary characters (X'00' to X'FF')

Sequence field position
The position (starting with 1) of the sequence field within the output block or record. If omitted, there is no sequence field.

This field is ignored if a template or copybook is used for data creation.

Note:

1. This differs from the Key position field on the Allocation panel, which measures the position as an offset value. That is, if the Allocation panel defines the Key position as 10, then the field begins in column 11.
2. **Key position** and **Key length** define a unique sequencing field located anywhere in the record, and not necessarily related to the key of a KSDS VSAM data set, as they can be used with any type of data sets.

Sequence field length
The length of the sequence field. The default is 8. The value of [**Key position** + **Key length** - 1] must be less than or equal to the record length.

This field is ignored if a template or copybook is used for data creation.

Sequence field increment
The increment value used to increase the number in the sequence field. The default is 10.

This field is ignored if a template or copybook is used for data creation.

Copybook or Template
Data set name and Member name of the template or copybook to be used to format your data.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Disposition
Sets the output data set status:

Old/Reuse
Writes records into the output data set, starting from the beginning of the set and replacing any existing records.

Mod Appends the records to the end of the output data set.

Like data set
Name of the data set to be used as the model for the output allocation.

File Manager uses the TSO prefix (as defined for your user ID) used as the high-level qualifier for any name that is not enclosed in quotes.

If copying data sets, "=" denotes the current input data set.

The model data set must reside on a DASD volume, be cataloged, and SMS must be active.

Copybook or template
Determines whether or not a copybook or template is to be used to format your data.

Above File Manager uses the template specified on the panel. (If you specified a copybook, File Manager compiles it into a template before use.)

Previous
File Manager uses the last template you used to process this data set. If you have not previously used a template for this data set, File Manager displays a message and processing continues without a template.

None File Manager does not use a template and therefore does not perform data creation at field level.

Create dynamic
File Manager creates a dynamic template. You can now define fields and field creation attributes.

Edit template
Allows you to edit the template you have requested for processing.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Batch execution
Allows you to edit the JCL to run the function in batch.

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The field only displays if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. An I/O exit can only be used to process the data set in which you are creating records. It cannot be used to process the copybook or template that you are using to format the data set.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

ISPF Packing
Determines whether or not the output records are created in ISPF PACK format.

1 None
The output data set is written without ISPF packing.

2 Pack

Parent panels

- "Utility Functions menu panel" on page 703

Child panels

- "Data Set Selection panel" on page 519
- "Member Selection panel" on page 600
- "Dynamic Template panel" on page 529
- "Field Selection/Edit panel" on page 565
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

- "DSG (Data Set Generate)" on page 922

Related tasks and examples

- "Creating data sets and records" on page 242

Data Set Selection panel

The Data Set Selection panel lists all data sets that match the data set name pattern provided on a process Entry panel, for example, on the View Entry panel. From this panel, you can select a data set for processing.

The panel can be called from a number of different locations (see "Parent panels" below). On some occasions, the panel title is modified to reflect its current usage. For example, when called from the Copy Utility From panel, the title reads "Copy From Data Set Selection".

Panel and field definitions

```
 Process   Options   Help
 File Manager               Data Set Selection            Row 00001 of 00006
Data Set Selection for FMNUSER.FMDATA.KSD*
  Data Set Name                                    Type  Volume Created
  TYRONED.FMDATA.KSD*                              *     *      *
_ FMNUSER.FMDATA.KSDS                              KSDS         2007.242
_ FMNUSER.FMDATA.KSDS.DATA                         DATA  D$US57 2007.242
_ FMNUSER.FMDATA.KSDS.INDEX                        INDEX D$US57 2007.242
_ FMNUSER.FMDATA.KSDS2                             KSDS         2008.352
_ FMNUSER.FMDATA.KSDS2.DATA                        DATA  D$US59 2008.352
_ FMNUSER.FMDATA.KSDS2.INDEX                       INDEX D$US59 2008.352
_
****  End of data  ****









Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 152. Data Set Selection panel

Data Set Selection for
Indicates the pattern used on the Entry panel.

(Select field)
Where you can enter S to select your data set. While you can enter the S in more than one field, only the first selected data set is processed.

Data Set Name
Lists the data sets that match your pattern.

Type Lists the data set type for each data set displayed.

Volume
Lists the volume on which the data set is stored.

Created
Lists the creation data for each data set, in the format yyyy.ddd where ddd is the day number within the year (not the date). For example, 2003.036 is February 05, 2003.

Parent panels

- "AFP Print Browse panel" on page 442
- "Allocate panel" on page 448
- "Browse Entry panel" on page 459
- "Copy From panel" on page 491
- "Copy To panel" on page 497
- "Data Create Utility panel" on page 515
- "Compare Utility: "Old" and "New" panels" on page 471
- "Edit Entry panel" on page 541
- "Find/Change Utility panel" on page 574
- "Load Module Information panel" on page 592
- "Print Utility panel" on page 620
- "Template Workbench panel" on page 698

Child panels

- The display panel that would normally result from your starting Entry panel.

Related tasks and examples

- "Specifying a data set and a member name" on page 16
- "Manipulating your view of selection lists" on page 28

Delete Entry panel

The Delete Entry panel is used to delete catalog entries from the IDCAMS catalog system.

Panel and field definitions

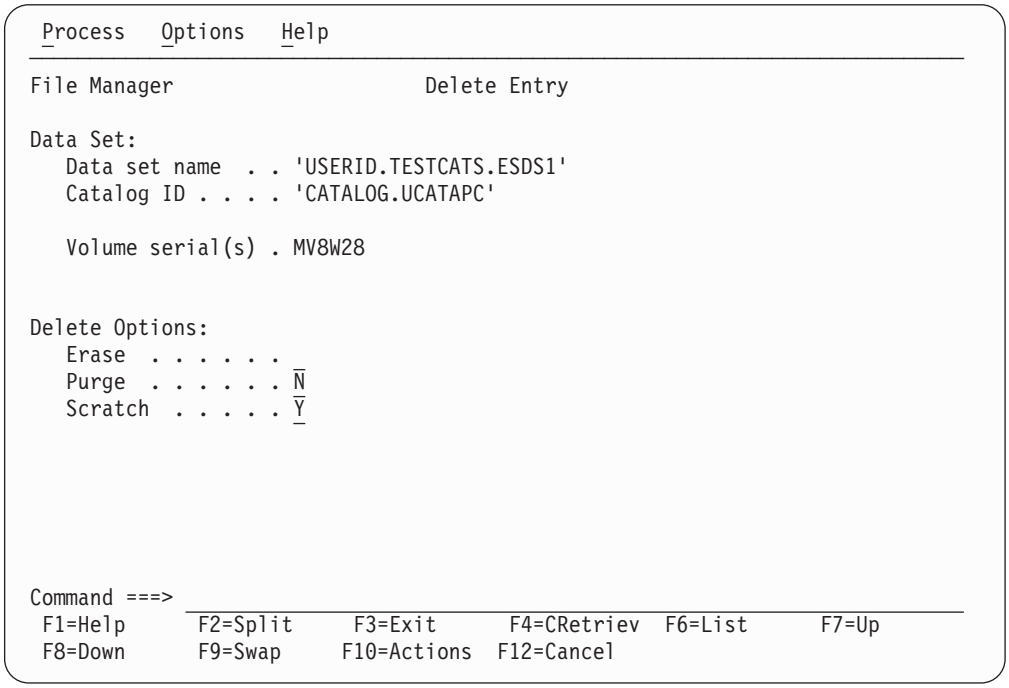


Figure 153. Delete Entry panel

The following options are available:

Erase Specify Y if the space occupied by the data set is to be overwritten with binary zeros when the data set is deleted; specify N if the space is not to be overwritten; or leave blank to use the erase option set when the catalog entry was defined or last altered.

Purge Specify Y to delete the entry regardless of the retention period specified, otherwise specify N.

Scratch

Specify Y if the data set is to be removed from the VTOC of the volume on which it resides, otherwise specify N.

This parameter is not applicable to entries which do not have a VTOC entry.

Parent panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Child panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Equivalent functions

- None

Related tasks and examples

- "Deleting a catalog entry" on page 323

Disk Volume Details panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                Disk Volume Details

Basic Information:                          Disk Geometry:
   Volser . . . . . D$US00                     Cylinders  . . . . . 10017
   Address  . . . . E903                       Tracks/cylinder  . . 15
   Device type  . . 3390-9                     Tracks total . . . . 150255
   Volstate . . . . PRIV/RSDNT                 Bytes/track  . . . . 58786
                                               Total capacity . . . 8423 MB

VTOC Information:                           Space Information:
   Start/end  . . . 00000.01-00007.14          Tracks used  . . . . 120725
   Size tracks  . . 119                        % used . . . . . . . 80
   % used . . . . . 11                         Tracks free  . . . . 29530
   Indexed  . . . . YES                        Max tracks free  . . 1246
                                               Cylinders free . . . 1835
Data Sets:                                     Max cylinders free . 82
   Total  . . . . . 615                        DSCBs free . . . . . 5282
   VSAM . . . . . . 30
   Non-VSAM . . . . 585

SMS Information:                            Storage Group Information:
   SMS managed  . . YES                        Storage Group type . POOL
   Storage group  . PRIMARY                    AUTO migrate . . . . YES
                                               AUTO backup  . . . . YES
                                               AUTO dump  . . . . . NO
                                               MIGR high  . . . . . 85
                                               MIGR low . . . . . . 02

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 154. Disk Volume Details panel

Parent panels

- "Volume Summary/Selection panel" on page 718

Child panels

- None

Equivalent functions

- None

Related tasks and examples

- "Displaying a Volume Table of Contents (VTOC)" on page 325

Disk/VSAM Data Functions panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager               Disk/VSAM Data Functions

1  Disk browse      Browse disk track data
2  Disk edit        Edit disk track data
3  Disk print       Print disk track data
4  Disk scan        Scan disk track data
5  Disk EOF         Write disk EOF record
6  Disk extents     Show single data set extents
7  VSAM update      Browse and update VSAM records or control intervals
8  Data In Virtual  Browse a VSAM data in virtual data set









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 155. Disk/VSAM Data Functions panel

Parent panels

"Primary Option Menu panel" on page 615

Child panels

- "Disk Browse panel"
- "Disk Track Edit panel"
- "Disk Print panel"
- "Disk Record Scan panel"
- "Write EOF Record panel"
- "Data Set Extents panel"
- "VSAM Update panel"
- "Data in Virtual Browse panel"

Related tasks and examples

- Chapter 9, "Disk/VSAM Data Functions," on page 371

Display VTOC panel

The Display VTOC panel is the entry panel for the Work with VTOC utility. In this panel, you can specify a data set name or name pattern, and the disk volume or volumes with which you want to work.

Panel and field definitions

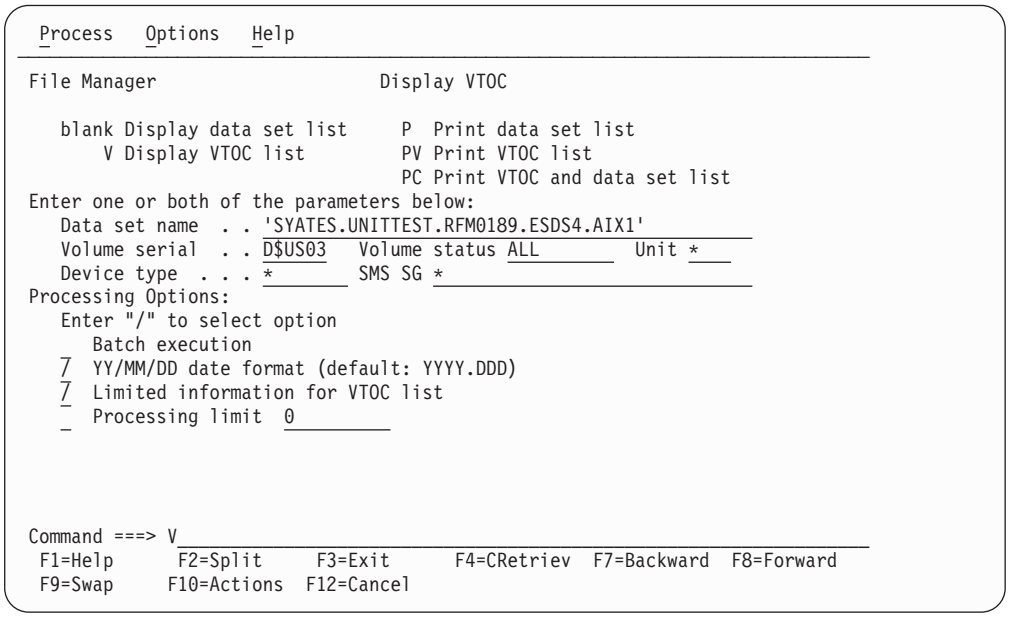


Figure 156. Work with VTOC panel

Data set name
Can be a fully-qualified data set name or a pattern. The data set name is not a required parameter. When no data set name has been entered, File Manager uses default generic data set name of ** which eventually resolves to 'userid.**' limiting the selected data sets to the files with names starting with the current userid.

Volume serial
A fully qualified or generic volume serial number.

In a generic volume use the percent % sign to represent a single character, and an asterisk * to represent any number of characters within one qualifier. Two asterisks ** represent any number of characters within any number of qualifiers.

Volume status
A fully-qualified volume status string (can be abbreviated). Can be one of the following values:
- PRIVATE (also includes PRIV/RSDNT)
- RESIDENT
- PUBLIC
- STORAGE
- ALL (default)

Unit A fully-qualified or generic device address (unit). In a generic name, use the percent sign (%) to represent a single character, and an asterisk (*) to represent any number characters within one qualifier.

Device type
A fully-qualified or generic device type (DEVTYPE). In a generic name, use the percent sign (%) to represent a single character, and an asterisk (*) to represent any number characters within one qualifier. For example, to list only volumes on 3390 devices, specify `3390`.

SMS SG
A fully-qualified or generic SMS storage group name. In a generic name,

use the percent sign (%) to represent a single character, and an asterisk (*) to represent any number characters within one qualifier.

In an SMS environment, the reserved name, NONSMS selects a non-SMS controlled volume.

Batch execution
Generates JCL to run the function in batch and then displays the JCL for editing. Use Set Batch Job Card Information (option 0.4) to tailor the default JOB card that File Manager uses to generate the JCL. See "Set batch job card information (option 0.4)" on page 45.

YY/MM/DD date format (default: YYYY.DDD)
Uses the YY/MM/DD date format instead of the default format of YYYY.DDD.

Limited information for VTOC list
If this option is selected, when you choose to display a VTOC list ("V" on the command line), File Manager does not collect data set statistics for the volumes processed and "N/A" is shown for the data set statistics on the Disk Volume Details panel. Select this option for better performance.

Processing limit
If this option is selected, the number of data sets selected for VTOC processing is limited to the number specified. The default of 0 allows for no limit on the number of data sets selected.

Parent panels

- "Utility Functions menu panel" on page 703
- "Volume Summary/Selection panel" on page 718

Child panels

- "Display VTOC Data Set List panel" on page 528
- "Volume Summary/Selection panel" on page 718

Related tasks and examples

- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Display VTOC Batch Sort Options panel"

Display VTOC Batch Sort Options panel

The Display VTOC Batch Sort Options panel displays the sort options available when the **Batch execution** option or **PV** line command have been selected from the Display VTOC panel.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager           Display VTOC Batch Sort Options

Sort data set list by           Sort volume list by

1  1. Name                      1  1. Volser
   2. Volser                       2. Device address (unit)
   3. Extent/Begin-end             3. Device type
   4. Size/Tracks                  4. SMS storage group
   5. Type/Dsorg                   5. Capacity/tracks
   6. Recfm                        6. Utilization/tracks
   7. Lrecl                        7. Utilization percentage
   8. Blksize                      8. Free tracks
   9. Creation date                9. Number of data sets
  10. Expiration date             10. Number of VSAM data sets
  11. Referred date               11. Number of non-VSAM data sets
                                  12. VTOC size
_  Repeat data set name           13. VTOC utilization
   for each extent                14. VTOC indexing
                                  15. Free DSCBs in VTOC
                                  16. Free space in cylinders
                                  17. Maximum available in cylinders
                                  18. Free space in tracks
                                  19. Maximum available in tracks

Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap      F10=Actions  F12=Cancel
```

Figure 157. Display VTOC Batch Sort Options panel

Sort data set list by
Specifies the sequence of the data sets in the Display VTOC Data Set List panel. Can be:
Name Default. Sorted by data set name
Volser Sorted by volume serial
Extent/Begin-end
Sorted by extent
Size/Tracks
Sorted by data set size
Type/Dsorg
Sorted by DSORG
Recfm Sorted by RECFM
Lrecl Sorted by LRECL
Blksize
Sorted by BLKSIZE
Creation date
Sorted by creation date
Expiration date
Sorted by expiration date
Referred date
Sorted by last referred date

Where an option other than Name is selected, the list is sorted first by the selected option and then by name.

Sort volume list by
Sorts the disks in a volume summary list. Can be:
Volser Sorted by volume serial
Device address (unit)
Sorted by device address (unit)

Device type
Sorted by device type
SMS storage group
Sorted by SMS storage group
Capacity/tracks
Sorted by disk capacity
Utilization/tracks
Sorted by tracks used
Utilization percentage
Sorted by percentage utilization
Free tracks
Sorted by tracks available
Number of data sets
Sorted by number of data sets
Number of VSAM data sets
Sorted by number of VSAM data sets
Number of non-VSAM data sets
Sorted by number of non-VSAM data sets
VTOC size
Sorted by VTOC size
VTOC utilization
Sorted by VTOC utilization in %
VTOC indexing
Sorted by VTOC indexing
Free DSCBs in VTOC
Sorted by VTOC free DSCB
Free space in cylinders
Sorted by cylinders available
Maximum available in cylinders
Sorted by maximum free space in cylinders
Free space in tracks
Sorted by tracks available
Maximum available in tracks
Sorted by maximum free space in tracks

The option takes effect only when the V (Display VTOC list), PV (Print VTOC list), or PC (Print VTOC and data set list) commands are executed.

Repeat data set name for each extent
Uses an alternate format for displaying or printing the data set name for multi-extent data sets. The default format suppresses the data set name on the second and subsequent extents. When you select this option, the data set name appears for each extent.

Parent panels

- "Display VTOC panel" on page 523

Child panels

None.

Related tasks and examples

- "Displaying a Volume Table of Contents (VTOC)" on page 325

Display VTOC Data Set List panel

The Display VTOC Data Set List panel shows the list of VTOC entries that match the information entered on the Display VTOC panel.

Panel and field definitions

```
 Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────────
 File Manager           Display VTOC Data Set List          Row 00001 of 00055
Unit    *        DSN       'FMN.V11R1M0.S*'
DevType *        VOLSTATE  ALL         SMS SG *
Volumes 10       Data sets 27          VSAM   0          non-VSAM 27
VOLSER  *        Trks used 3752        Free   N/A        Utilized N/A
        Data Set Name                            Seq Volume Begin CYL-HD  ±
        FMN.V11R1M0.S*                               *      *
_______ FMN.V11R1M0.SFMNCLIB                       1 D$FM08    32 14
_______ FMN.V11R1M0.SFMNDBRM                       1 D$FM09  8802  4
_______ FMN.V11R1M0.SFMNEXEC                       1 D$FM10  1060  0
_______ FMN.V11R1M0.SFMNMAC1                       1 D$FM00  7885  1
_______ FMN.V11R1M0.SFMNMENU                       1 D$FM19   172 13
_______ FMN.V11R1M0.SFMNMJPN                       1 D$FM09  9025  8
_______ FMN.V11R1M0.SFMNMKOR                       1 D$FM03  6363 10
_______ FMN.V11R1M0.SFMNMODA                       1 D$FM02  2881  0
_______ FMN.V11R1M0.SFMNMODA                       2 D$FM02   109  0
_______ FMN.V11R1M0.SFMNMODA                       3 D$FM02   109 10
_______ FMN.V11R1M0.SFMNMODA                       4 D$FM02   110  5
_______ FMN.V11R1M0.SFMNMODA                       5 D$FM02   111  0
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 158. Display VTOC Data Set List panel

Available commands

You can use the following primary command on this panel:

FIND Finds the next occurrence of a character string in the Data Set Name area. Allows for a string argument which might contain the * and % wildcard characters. The FIND argument is matched against the data set names using the default system match rules for such wildcard characters.

You can also use the following line commands in the prefix command area:

ALTER (A)
Invokes the Catalog Services Alter function which lets you change some data set attributes, such as expiration date and data set name (this functionality is only available for cataloged data sets).

BROWSE (B)
Invokes Browse (option 1).

CATALOG (CA)
Catalogs the data set (if not already cataloged).

COPY (C)
Copies the selected entry. Invokes the Copy Utility (3.3) with the “from” data set prefilled to the selected data set. If the input data set is cataloged, the volume information is not prefilled.

DELETE (D)
Deletes the data set, and uncatalogs it if cataloged.

EDIT (E)
Invokes Edit (option 2).

EXTENTS (EX)
Displays a pop-up panel showing the data set extents.

INFO (I)
Displays detailed information about the selected data set.

MEMBER (M)
If entered for a library data set entry, displays a member list.

PRINT (P)
Invokes Print Utility (option 3.2).

RENAME (REN)
This is a synonym for Alter with non-VSAM data sets and it works identically. For VSAM data sets it invokes a separate VSAM Entry Rename panel.

UNCAT (U)
Uncatalogs the data set (if cataloged).

Caution: uncataloging an SMS-managed data set might leave the data set in an unusable state.

VIEW (V)
Invokes AFP Print Browse (option 3.7).

Parent panels

- "Display VTOC panel" on page 523

Child panels

- None.

Equivalent functions

- "DVT (Display VTOC)" on page 1006

Related tasks and examples

- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "FIND/FX primary command" on page 769
- "RFIND primary command" on page 809
- "LOCATE primary command" on page 786
- "REFRESH primary command" on page 804
- "SORT primary command" on page 818
- "Manipulating your view of selection lists" on page 28

Dynamic Template panel

The Dynamic Template panel is used to define fields and set record selection criteria within a dynamic template (see "Creating dynamic templates" on page 152).

Panel and field definitions

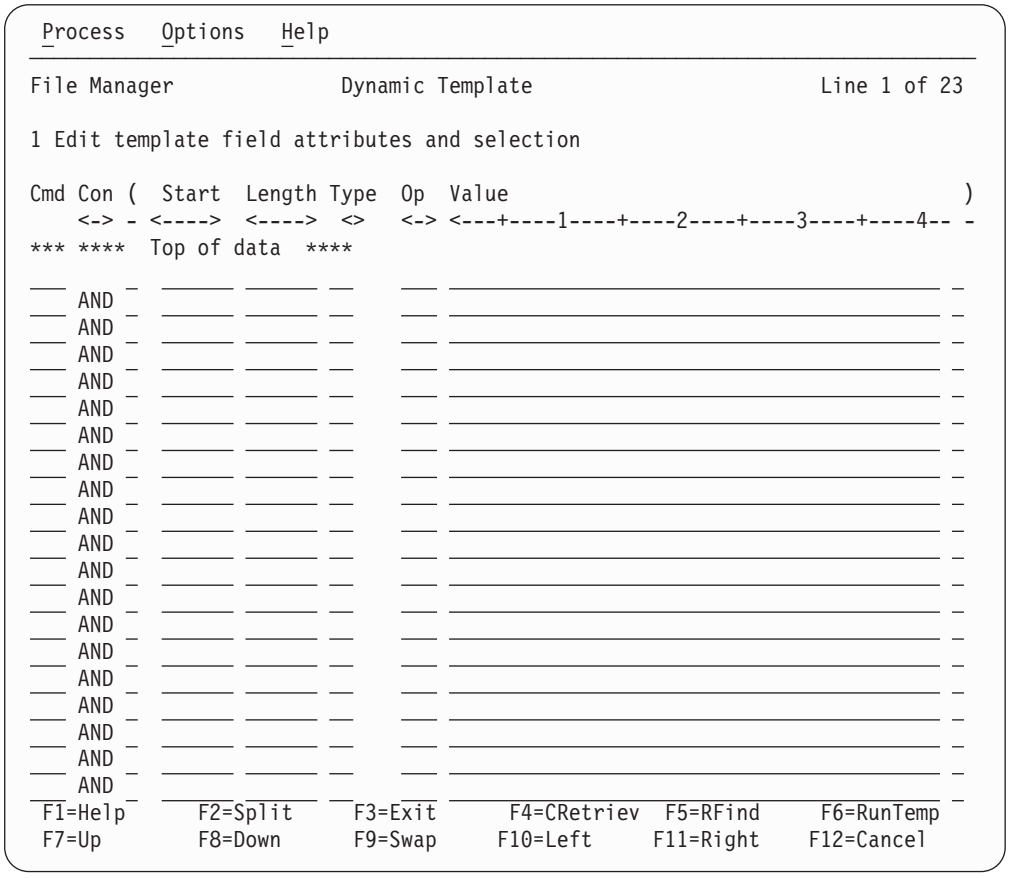


Figure 159. Dynamic Template panel

Cmd Prefix command area - used to enter a template editor prefix command.

Con Connectors - used to specify the AND/OR connectors. These are only relevant to lines containing record selection expressions. The default value is **AND**. You can overtype the connectors with any of the following values:

Value entered	Resultant value
AND	AND
&	AND
OR	OR
|	OR

Note: Abbreviations are accepted. For example, overtype `AND` with `O` and it is changed to `OR`.

(Left Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record selection criteria. The number of left parentheses must match the number of right parentheses.

Start An editable field in which you can enter or edit a field's starting column number. This must be between 1 and the record length or, if you specify a + sign optionally followed by an offset number (default zero), the start location is calculated as:

```
(previous field start) + (previous field length) + offset
```

For example, +0 starts at the next contiguous location and +4 starts 4 bytes past the next contiguous location.

Length

An editable field in which you can enter or edit the length of a field. For most fields, this can be from 1 to the maximum length dictated by the record but the Start value plus the Length of the field should not exceed the record length. Restrictions apply for some field types:

If the Type is:	then Length must be:
BI	2, 4 or 8
FE	> 6 and < 24. The precision is the length minus 6. The scale is the precision minus 1. Graphic and DBCS strings must be an even number of bytes.
FP	4 or 8
PD	between 1 and 16
ZD	between 1 and 32

Type An editable field in which you can enter or edit the data type of a field. This must be one of:

AN Alphanumeric

AX Alphanumeric displayed in long hexadecimal.

B Binary

BI Binary

BT Bit string

C Alphanumeric

DB DBCS string

FE Floating point (external)

FP Floating point (internal)

G Graphic string

P Packed decimal

PD Packed decimal

VB Varying bit

VC Varying character

VD Varying DBCS

VG Varying graphic

Z2 Null-terminated varying-length(VARYINGZ) DBCS string

ZA Unsupported zoned decimal edited

ZC Varying character null terminated

ZD Zoned decimal

ZG Varying graphic null terminated

Field Name

An editable field in which you can enter or edit the field names in your dynamic template. The name can consist of any characters, including spaces, up to a maximum of 35.

If the Start, Length and Type fields are filled in but the Field Name is left blank, a name is generated in the form `@@DT`*nn* where *nn* is the field number. The name is generated either on exit or when you perform a template edit. Existing field names taking the form `@@DT`*nn* are regenerated to ensure *nn* always reflects the current field number.

Note: If you are running a screen width of 115 or greater, the Field Name field can be displayed at the same time as the Start, Length and Type fields. However, at any width less than this, you need to toggle the display between Field Name and Start, Length and Type. The default is to display Start, Length and Type. To toggle the display, ensure that your cursor is not in a Value field and press the Right function key (F10) or Left function key (F11).

Op Operator - used to create record selection expressions. Valid operators are:

=, EQ True if the terms are equal (numerically or when padded)

\=, ¬=, /=, NE
True if the terms are not equal (inverse of =)

>, GT Greater than

<, LT Less than

><, <> Greater than or less than (same as not equal)

>=, GE
Greater than or equal to

\<, ¬< Not less than

<=, LE Less than or equal to

\>, ¬> Not greater than

== True if terms are strictly equal (identical)

\==, ¬==, /==
True if the terms are NOT strictly equal (inverse of ==)

>> Strictly greater than

<< Strictly less than

>>= Strictly greater than or equal to

\<<, ¬<<
Strictly NOT less than

<<= Strictly less than or equal to

ACO Contains all of the values specified. Matching case sensitive. The negative form is XCO.

ACU Contains all of the values specified. Matching not case sensitive. The negative form is XCU.

CO Contains one of the values specified. The negative form is ¬CO.

CU Contains one of the values specified. Matching not case sensitive. The negative form is ¬NC.

NC Numeric contains - a number is one of the values. The negative form is ¬NC.

NN (Same as ¬NU.) Field does not contain a valid numeric for the field type. The positive form is NU.

NR Not in range - field < value1 or field > value2. The positive form is RG.

NU Field contains a valid numeric for the field type:

Field type
Test for being valid numeric

BI, FE, FP, PD, and ZD
Checked against the field definition

Other Recognized as numeric if the display value or character value the field contains is a valid number. The valid number can contain anything that is valid for a floating-point external or decimal number. For example:

```
-1234.55
+3.4621E+10
```

The negative form is ¬NU.

RG Range - value1 >= field <= value2. The negative form is NR.

XCO Does not contain all of the values. Matching case sensitive. The positive form is ACO.

XCU Does not contain all of the values specified. Matching case sensitive. The positive form is ACU.

¬CO Does not contain one of the values specified. The positive form is CO.

¬CU Does not contain any of the values specified. Matching not case sensitive. The positive form is CU.

¬NC Does not contain the specified number(s). The positive form is NC.

¬NU (Same as NN.) Field does not contain a valid numeric for the field type. The positive form is NU.

VER Verify this field is composed only of characters specified in the **Value** column.

Value Use this field to provide a character or numeric constant to be used for comparison in the expression. If the operator supports more than one value (for example, RG, CO) then use comma delimiters to provide separate values.

You can use up to 255 characters in this field, and use the Left and Right primary commands or function keys to scroll within the field.

When specifying values, the following rules apply:

Hexadecimal strings
A hexadecimal string must be in the form 'hhhhhh'x or x'hhhhhh'. The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters. (0123456789ABCDEF).

Binary strings
A binary string must be in the form 'nnnnnn'b or b'nnnnnn'. The value enclosed in quotes must be a combination of 0 and 1s.

Numeric Operands
The value(s) specified for a numeric field type must be a valid number. Quoted values are not accepted.

Character strings
For non-numeric types the value is automatically enclosed in quotes if you do not specify them. When specifying more than one value for operations other than RG and NR you must provide quotes for all the comma delimited values

) Right Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record selection criteria. The number of left parentheses must match the number of right parentheses.

Parent panels

- "Browse panel" on page 455
- "Browse Entry panel" on page 459
- "Editor panel"
- "Edit Entry panel" on page 541
- "Compare Utility: "Old" and "New" panels" on page 471
- "Copy From panel" on page 491
- "Copy To panel" on page 497
- "Data Create Utility panel" on page 515
- "Find/Change Utility panel" on page 574
- "Print Utility panel" on page 620
- "Template Workbench panel" on page 698

Child panels

- The display panel that would normally result from your starting Entry panel.

Equivalent functions

- None.

Related tasks and examples

- "Creating dynamic templates" on page 152

Editor panel

You use the Editor panel to display a selected data set or data set member, scroll through the records and find and change specific information within the records. You can also use this panel to insert or delete new records, join or split existing records and copy or repeat records within the data set.

The Editor panel displays different fields, depending upon the type of data set shown and whether or not a template has been used.

Panel and field definitions

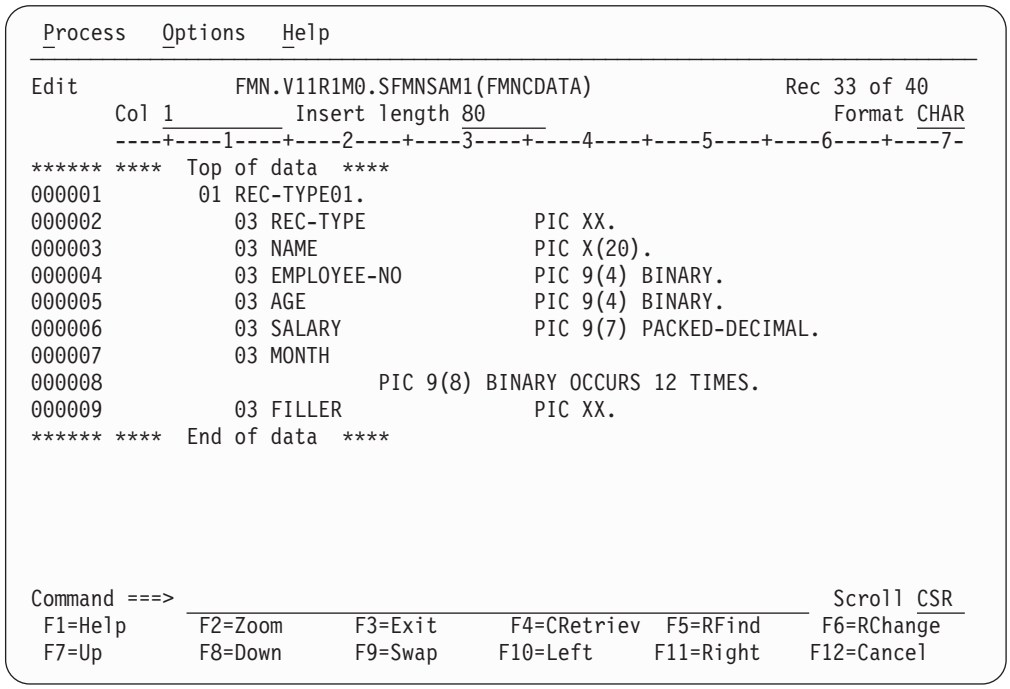


Figure 160. Editor panel showing text file

Title The Title identifies the function (Edit) and the data set being used. For a PDS or PDSE member, the data set name includes the member name. Record number information, or short messages, appear to the right.

Col The Col field shows the column number of the column that is currently at the far left of the data area. You can scroll to a specific column (left or right) by typing a new value. The default value for this field is 1.

Insert length
The Insert length field specifies the initial length of inserted records. The possible range of values is from 1 to 9999999, depending on the data set characteristics. The default value is the maximum record length as defined in the catalog. When a User I/O exit is being used, the insert length value is adjusted to the maximum length accepted by the exit for the data set being edited.

Scale The Scale shows the columns of the data area.

Prefix area (line numbers)
Area (if displayed) displaying the record number of each line in the data set. This area also doubles as the prefix command entry area.

You can use the following prefix commands when editing:

A Identifies the record after which data is to be moved or copied.

A "K" can be appended to the A to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

B Identifies the record before which data is to be moved or copied.

A "K" can be appended to the B to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

BND (Available only in CHAR, HEX and LHEX display formats.) Displays a bounds line on the following line, indicating the columns searched by the CHANGE, FIND, and EXCLUDE commands. To change the bounds, type a < on the bounds line to define the left bound and a > to define the right bound. To remove the bounds line from the display, use the D prefix command or the RESET SPECIAL primary command.

C Copy one record.

C*n* Copy *n* records.

CC Copy block of records. Mark start and end of block.

COLs Displays the column identification line (CHAR, HEX, and LHEX display formats only).

D Delete one record.

D*n* Delete *n* records.

DD Delete block of records. Mark start and end of block.

F Display the first record of a block of excluded records.

FC Changes the display format to CHAR.

FH Changes the display format to HEX.

FL Changes the display format to LHEX.

FS Changes the display format to SNGL.

FT Changes the display format to TABL.

F*n* Display the first *n* records of a block of excluded records.

I Insert one empty record.

If the display format is CHAR, HEX or LHEX, then the entire record is initialized to blanks.

If the display format is TABL, then the record is initialized according to the record structure of the current record type:

- Numeric fields are initialized to zero.
- Alphanumeric fields are initialized to blanks.
- If the record type contains a variable-length array, then the record is initialized with the minimum number of array elements, and the array size field (or fields, for multi-dimensional arrays) is initialized accordingly.

I*n* Insert *n* empty records.

L Display the last record of a block of excluded records.

L*n* Display the last *n* records of a block of excluded records.

LC Translate all uppercase characters in a record contents to lower case (see **Caution** in LCC).

LC*n* Translate all uppercase characters in *n* records to lower case (see **Caution** in LCC).

LCC Translate all uppercase characters in a block of records to lower case. Mark start and end of block.

Caution: The LC commands (LC, LC*n*, LCC) affect all characters in a record, not just characters in those fields with an alphanumeric or character data type. This means that numeric data, such as binary data or packed decimal data, can be corrupted by using these commands.

M Move one record.

M*n* Move *n* records.

MM Move block of records. Mark start and end of the block.

O The O (overlay) command indicates the target when you want to merge lines. Overlay is used in conjunction with move and copy commands.

A "K" can be appended to the O to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

O Overlay one record.

O*n* Overlay *n* records.

OO Overlay block of records. Mark start and end of the block.

P Identifies the record preceding which data is to be moved or copied. This is synonymous with the B prefix command.

R or " Repeat one record.

R*n* **or "***n*
Repeat *n* records.

RF Perform a refresh of the record. (Shared files only.)

RF*n* Perform a refresh of *n* records. (Shared files only.)

RFF Perform a refresh of a block of records. Mark start and end of the block. (Shared files only.)

RR or ""
Repeat block of records. Mark start and end of the block.

RR*n* **or ""***n*
Repeat block of records *n* times. Mark start and end of block.

SV Perform a save of the record. (Shared files only.)

SV*n* Perform a save of *n* records. (Shared files only.)

SVV Perform a save of a block of records. Mark start and end of the block. (Shared files only.)

UC Translate all lowercase characters in a record to upper case (see **Caution** in UCC).

UC*n* Translate all lowercase characters in *n* records to upper case (see **Caution** in UCC).

UCC Translate all lower case characters in a block of records to upper case. Mark start and end of block.

Caution: The UC commands (UC, UC*n*, UCC) affect all characters in a record, not just characters in those fields with an alphanumeric or character data type. This means that numeric data, such as binary data or packed decimal data, can be corrupted by using these commands.

V (Available only when you are using a template.) Display currently suppressed records of the record type indicated by the shadow line. Records of other record types are suppressed from display.

X Exclude record from display. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position. To show an excluded record, use the F or LA prefix command.

X*n* Exclude *n* records from display. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position.

XX Exclude block of records from display. Mark start and end of block. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position.

.*xxxx* Label a line. The label, *xxxx*, is a string of 1 to 4 alphabetic characters, that does not start with the character "Z" (labels starting with "Z" are system labels). Labels can be assigned to any data line. Labels cannot be assigned to the bounds line or to shadow lines.

) Shift record right one position.[1]

)*n* Shift record right *n* positions.[1]

)) Shift block of records right one position. Mark start and end of block.[1]

))*n* Shift block of records right *n* positions. Mark start and end of the block.[1]

(Shift record left one position.[1]

(*n* Shift record left *n* positions.[1]

((Shift block of records left one position. Mark start and end of block.[1]

((*n* Shift block of records left *n* positions. Mark start and end of block.[1]

/ Position record at the top of the screen.

1. The shift occurs only for that part of the record within the current bounds limit. See "Setting bounds" on page 109 for a detailed explanation.

Data Area
The Data Area shows the data in the selected display format. For a description of the different display formats, see "Selecting a display format" on page 66.

Command
The Command line is a field in which you can enter Primary Commands, such as CHANGE.

Scroll The Scroll field defines the current scroll amount. You can type a new value.

When a VSAM data set is displayed, additional fields can be seen.

```
Edit            FMN.REQ77.RRDS                                    Rec 0
       Type RRDS                                                Format CHAR
       Col 1________ Insert length 4089
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
****** ****  Top of data  ****
000001 xxxxxxyyy orig 1
```

Figure 161. Edit panel showing VSAM data set

Type The type of VSAM data set, for example, RRDS. IAM files are also indicated.

When a data set is displayed in TABL format with a template, the column headings show the field names defined in the template.

```
Edit            FMN.V11R1M0.SFMNSAM1(FMNCDATA)              Rec 13 of 40
                                                            Format TABL
       REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY  MONTH(1)
       #2       #3                         #4       #5       #6        #7
       AN 1:2   AN 3:20               BI 23:2 BI 25:2  PD 27:4   BI 31:4
       <>       <---+----1----+---->      <---+>  <---+> <---+--> <---+--->
****** ****  Top of data  ****
000001 01       GRAHAM JONES               5512      94    68000         7
```

Figure 162. PDSE data set member in TABL format

When a data set is displayed in SNGL format with a template, the field and record number of the current record is shown.

```
Edit            FMN.V11R1M0.SFMNSAM1(FMNCDATA)              Rec 13 of 40
Current type is REC-TYPE01                                      Format SNGL
                                 Top Line is 1    of 32    in Record 3
Ref Field    Picture Typ Start   Len   Data
  2 REC-TYPE
              XX      AN      1     2   01
  3 NAME      X(20)   AN      3    20   Graham Jones
  4 EMPLOYEE-NO
              9(4)    BI     23     2    5512
  5 AGE       9(4)    BI     25     2    94
  6 SALARY    9(7)    PD     27     4    68000
```

Figure 163. PDSE data set in SNGL format

Available commands

- "TOP primary command" on page 824
- "TVIEW primary command" on page 825
- "TYPE primary command" on page 826
- "UP primary command" on page 826
- "VIEW primary command" on page 828
- "ZOOM primary command" on page 829

Parent panels

- "Browse Entry panel" on page 459

Child panels

- "Template Workbench panel" on page 698 (to invoke the panel, enter the TVIEW primary command).
- "Record Type Selection panel" on page 636 (to invoke the panel, enter the TEDIT primary command - panel displays when template contains more than one record type).
- "Field Selection/Edit panel" on page 565 (to invoke the panel, enter the TEDIT primary command - panel displays when using a copybook template with only one record type).
- "Dynamic Template panel" on page 529 (to invoke the panel, enter the TEDIT primary command - panel displays when using a dynamic template).
- "Record Selection Criteria panel" on page 631 (to invoke the panel, enter the CEDIT primary command).
- "Record Identification Criteria panel" on page 625 (to invoke the panel, enter the CEDIT ID primary command).

Equivalent functions

- "DSE (Data Set Edit)" on page 900
- "DSEB (Data Set Edit Batch) — batch only" on page 907
- "DSU (Data Set Update) — batch only" on page 984

Related tasks and examples

- Chapter 3, "Viewing and changing data sets," on page 49

Edit Entry panel

The Edit Entry panel is virtually identical to the View Entry panel with two additional options, **Inplace edit** and **Create audit trail**. For an explanation of the fields in this panel, see the field definitions in "Browse Entry panel" on page 459.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Edit Entry Panel

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name 'FMNUSER.EXPORT'                                  +
   Member . . . . . _________     Blank or pattern for member list
   Volume serial  . ______        If not cataloged
   Start position . ______________________________ +
   Record limit . . _________     Record Sampling _
   Inplace edit . . _               (Prevent inserts and deletes)
Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . _________   Blank or pattern for member list

Processing Options:
 Copybook/template   Start position type   Enter "/" to select option
 1  1. Above         _  1. Key             _  Edit template   _ Type (1,2,S)
    2. Previous         2. RBA             _  Include only selected records
    3. None             3. Record number   _  Binary mode, reclen _____
    4. Create dynamic   4. Formatted key   _  Create audit trail
                                           _  Use I/O exit
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 164. Edit Entry panel

Inplace edit
If you select this option, the edit session does not allow you to insert or delete records.

If you deselect this option, then for QSAM, VSAM ESDS and VSAM RRDS data sets that are larger than the available virtual storage, or when using this in conjunction with a start position or record limit, File Manager uses an auxiliary file in order to allow rewriting of the data set.

Inplace edit is implied when using any of the following:
- A segmented record type data set (as indicated by your chosen template),
- Record sampling,
- A record limit of “MEMORY”, or
- A VSAM data set defined as NOREUSE

Create audit trail
Determines if File Manager generates an audit report of all successful modifications to data made during an Edit session.

The display of this option depends on whether SAF-rule controlled is in effect. See “SAF-rule controlled auditing” on page 47.

The ability to change this option depends on the installation options (in the FMN0POPT macro).

When your system administrator has enforced audit logging, the **Create audit trail** option in this panel is ignored.

When your system administrator has not enforced audit logging, you can set this option on or off for any particular editing task. Type a “/” in the option entry field to generate audit reporting for the current Edit session.

For more information about setting the Audit Trail options, refer to the *File Manager Customization Guide* or see your systems administrator.

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

Depending upon the options selected in the Edit Entry panel, the next panel displayed can be:

- "Editor panel" on page 534
- "Data Set Selection panel" on page 519 (a pattern has been entered in the Data set name field)
- "Member Selection panel" on page 600(a pattern or a blank has been entered in the Member field)
- "Record Type Selection panel" on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- "Field Selection/Edit panel" on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).
- "Dynamic Template panel" on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

- "DSC (Data Set Copy)" on page 875

Related tasks and examples

- "Browse Entry panel" on page 459
- "Copying data sets" on page 253
- "Supplying a procedure when using a File Manager panel" on page 417

Edit Personal Data Set List panel

The Edit Personal Data Set List panel displays the contents of a data set list and allows you to edit it by changing or deleting existing entries or adding more entries.

Panel and field definitions

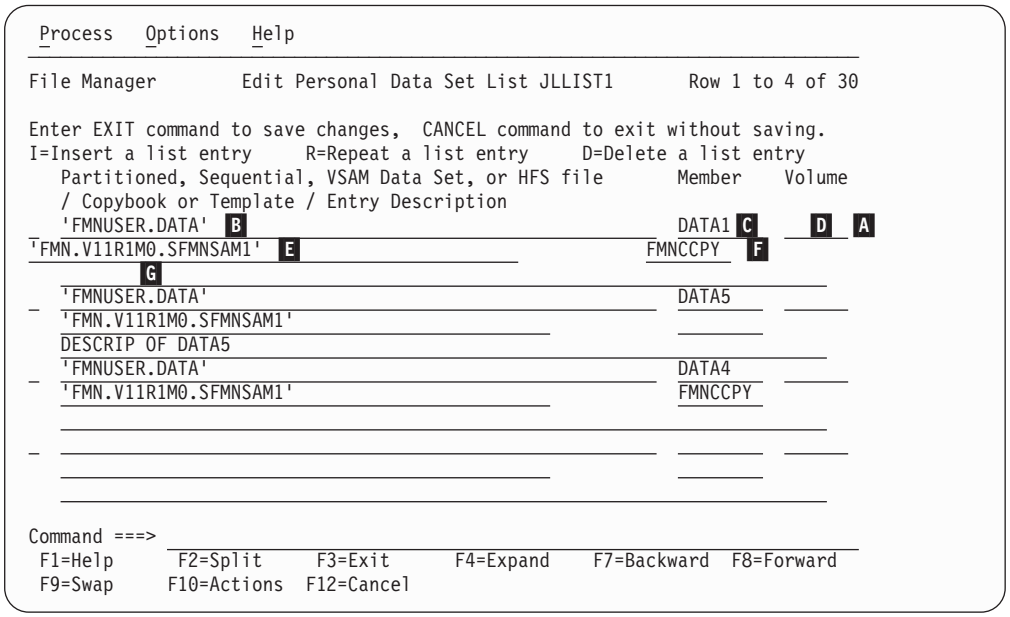


Figure 165. Edit Personal Data Set List panel

A Line command entry field. Valid values are:
- **I** Insert a list entry
- **R** Repeat a list entry
- **D** Delete a list entry

B Data set name

C Member name

D Volume serial number

E Data set name of copybook or template

F Member name of copybook or template

G Description line for entry

Parent panels

- "Personal Data Set List panel" on page 612
- "Personal Data Set Lists panel" on page 614

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Working with data set lists" on page 235

Edit/View - Copy panel

The Edit/View - Copy panel allows you to specify details about the data you want to copy into a data set.

File Manager displays this panel when you enter the COPY primary command in a browse or edit session but without specifying the data set or member of a partioned data set from which the data is to be copied.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Edit/View - Copy

From Data set:
   Data set/path name  . . 'JOHNLEV.TEST.DATA'                          +
   Member name (or mask) . ________
   Volume serial . . . . . ______

Line Numbers (Blank for entire member or data set)
   First line  . . . . . . ________
   Last line . . . . . . . ________

Press Enter key to copy, enter End command to cancel copy.

Processing Options:
Enter anything to select option
_  Binary mode, reclen _____



Command ===>
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 166. Edit/View - Copy panel

Data set/path name
Can be a fully-qualified data set name or a pattern, or a HFS file or directory. The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member name (or mask)** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Member name (or mask)
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set. Required for data sets which are not cataloged.

First line
The number of the first line in the source file from which you want to copy data.

Last line
The number of the last line in the source file from which you want to copy data.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Parent panels

- "Browse Entry panel" on page 459
- "Editor panel" on page 534

Related tasks and examples

- "Copying data from another data set" on page 122

Editor Options panel

The editor options control various options used to format the display or print of data when viewing, editing, or printing.

Note: The editor options are displayed on 3 panels ("Editor Options (1 of 3)" through to "Editor Options (3 of 3)").

To navigate from one panel to the next, press F11 (NxtPage) or F10 (PrvPage).

For simplicity, the contents of all 3 panels are shown together in Figure 167 on page 547.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager              Editor Options (all 3 panels combined)

Related command if applicable, shown in ()
Initial Display
1  1. Previous
   2. Table
   3. Single
   4. Character
   5. Hex
   6. LHex

Miscellaneous:     Enter "/" to select option
/  Recognize and interpret ISPF packed data
_  CAPS initially ON - translate changed data to uppercase (CAPS)


Related command if applicable, shown in ()
Grouping and Hiding options:     Enter "/" to select option
Expose (do not group) records of types:            (SHOW)
/  Not selected  /  Suppressed  /  Length error
See shadow lines (deselect to hide) for groups of: (SHADOW)
/  Not selected  /  Suppressed  /  Excluded

Multi-line display format options:  (TABL/CHAR/HEX/LHEX)
/  Display prefix area                             (PREFIX on/off)
_  Display prefix area on the right                (PREFIX RIGHT)
A  Prefix area width  (A,6,7,8,9)                  (PREFIX _)
_  Display record length                           (RECL on/off)
_  Display record length on the right              (RECL RIGHT)
A  Record length width  (A,3,4,5,6)                (RECL _)
_  Display RBA and Length when browsing VSAM       (RBALEN on/off)


Related command if applicable, shown in ()
Formatted display options:
/  Field reference number (REF) /  Field type and length values (TYPE)
_  Picture clause         (PIC) _  Start location               (SLOC)
_  Structure              (STR) _  Left justify numeric fields  (JUST)
/  Redefined fields       (RDF) /  CCSID                       (CCSID)


Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F10=PrvPage  F11=NxtPage  F12=Cancel
```

Figure 167. Editor Options panel

Initial Display

1. Previous
: At the start of an editor session, the display format field (**Format**) on the View or Edit panel is pre-filled with the value specified in the previous editor session. This is default.

2. Table
: At the start of an editor session, the View or Edit panel is displayed in TABL display format.

  For data that has not been formatted with a template, the View or Edit panel is displayed in CHAR display format.

3. Single
: At the start of an editor session, the View or Edit panel is displayed in SNGL display format.

For data that has not been formatted with a template, the View or Edit panel is displayed in CHAR display format.

4. Character
At the start of an editor session, the View or Edit panel is displayed in CHAR display format.

5. Hex At the start of an editor session, the View or Edit panel is displayed in HEX display format.

6. LHex
At the start of an editor session, the View or Edit panel is displayed in LHEX display format.

Recognize and interpret ISPF packed data
When this option is selected, File Manager checks the data in the data set or member being edited (or viewed) to determine if it has been written with the ISPF PACK option. If it had, File Manager then unpacks the data to allow it to be edited or viewed in the normal fashion. If the data set is too large to be contained in memory, File Manager cannot edit or view the data set in unpacked form, but instead provides the data set in its packed form. If this option is deselected, File Manager does not check the data and edit or view operates on the record data in its "as is" state.

Note: This option allows you to work with packed data in PDS or PDSE data sets. File Manager does not support working with packed data in VSAM data sets.

CAPS initially ON - translate changed data to uppercase (CAPS)
Translates data entered in input fields into uppercase.

Expose (do not group) records of types:

Not selected
Exposes or shows not-selected records. Selecting this option has the same effect as if the SHOW NOT ON command has been entered. If a SHOW command is entered during a File Manager session, it resets this option for the session.

Suppressed
Exposes or shows suppressed records. Selecting this option has the same effect as if the SHOW SUP ON command has been entered. If a SHOW command is entered during a File Manager session, it resets this option for the session.

Length error
If a record matches the record identification criteria for a record type in the template, but the record length is outside the valid range for the record type, then match the record with that type anyway.

If no record identification criteria are defined, and the length of a record does not fall within the valid range of any record type in the template, then match the record with the first record type in the template anyway.

In Edit, display `=LGTH` in the record prefix area.

If you do not select this option, then records whose length do not match a record type in the template are not selected, even if they match the record identification criteria for a record type.

There is no equivalent primary command for this option.

See shadow lines (deselect to hide) for groups of:

Not selected
Show shadow lines for not-selected records. Equivalent to the SHADOW NOT ON primary command.

For a definition of not-selected records, see "Filtering record display using templates" on page 224.

Suppressed
Show shadow lines for suppressed records. Equivalent to the SHADOW SUP ON primary command.

For a definition of suppressed records, see "Filtering record display using templates" on page 224.

Excluded
(Edit only.) Show shadow lines for excluded records (records that you have excluded from display using the X prefix command or the EXCLUDE primary command). Equivalent to the SHADOW EX ON primary command.

Multi-line display format options: (TABL/CHAR/HEX/LHEX)

Display prefix area
(Edit only.) Display the prefix area. Equivalent to the PREFIX ON or PREFIX OFF primary commands.

Display prefix area on the right
Place the prefix area (if it is displayed; see **Display prefix area**, above) on the right of the Edit panel. Otherwise, place the prefix area on the left. Equivalent to the PREFIX RIGHT primary command.

Prefix area width (A,6,7,8,9)
Sets the length of the prefix area displayed in the editor session. The prefix length can be from 6 to 9, or you can set it to A to automatically adjust the prefix area length as required in order to display the record numbers (when applicable) for your data set. A is the default value. You can also control the display and positioning of the prefix area with the PREFIX primary command.

Display record length
Turns on the display of record lengths in multiple line formats (TABL,CHAR,HEX,LHEX). If the **Display record length on the right** option is not selected, displays the record length on the left side of the panel. Enter any character to enable the display, or remove or overtype with a blank to disable. You can also control the display of the record length with the RECLEN primary command.

This option also controls whether the record length is printed when printing TABL and SNGL formatted records.

Display record length on the right
Displays the record length area in the File Manager Edit data display on the right side of the panel. If this option is not selected and **Display record length** is selected, the record length area is on the left side of the panel. This option is only effective when either the **Display record length** option is selected, or when you issue

the RECLEN ON command during an editor session. You can also control the display and positioning of the record length area with the RECLEN primary command.

Record length width (A,3,4,5,6)
In multiple line formats (TABL,CHAR,HEX,LHEX), sets the width of the record length area displayed in the editor session. You can set the record length width from 3 to 6, or you can set it to A to automatically adjust the record length area width as required in order to display the record lengths for your data set. A is the default value. You can also control the display and positioning of the record length area with the RECLEN primary command.

Display RBA and Length when browsing VSAM
When you browse a VSAM file, the leftmost side of the multiline displays contain the RBA and the length of the record being displayed.

If you deselect this option, File Manager does not display this information, allowing more of the record data to be displayed on each panel.

Formatted display options:

Field reference number (REF)
Changes the session default to show the field reference (for example, #2) next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the REFS command.

Field type and length values (TYPE)
Changes the session default to show the field data type and length next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the TYPE command.

Picture clause (PIC)
Changes the session default to show the Picture clause when in SNGL display or print mode. Also shows the length and scale (if non zero) for binary and packed fields and the bit length for bit fields. In an editor session, this display can also be turned on or off temporarily, by using the PIC command.

Start location (SLOC)
Changes the session default to show the start location of each field, when in SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the SLOC command.

Structure (STR)
Changes the session default to show the full structure of the template when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the STR command.

Left justify numeric fields (JUST)
Changes the session default to show numeric fields as left justified, when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the JUST command.

Redefined fields (RDF)
Changes the session default to show the redefined information as part of the field name, when in SNGL display or print mode and display or print the redefined fields when in SNGL or TABL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the RDF command.

CCSID (CCSID)
Change the session default to display the CCSID assigned to each field, if applicable, when in SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the CCSID command.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Printing from File Manager" on page 299
- "Viewing and changing packed data" on page 56
- "Setting the initial display format" on page 66

Erase Tape panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                        Erase Tape

Tapes:  No tapes allocated


Output:
   DDNAME to use . ________      enter new name, or select one from above
   Tape mode . . . __            optional recording mode or density code









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 168. Erase Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Exported Stacked Volume Copy panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager             Exported Stacked Volume Copy

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Logical vol(s)  ______________________________
                                 enter logical volume(s) to be copied

Output:
   DDNAME to use . ________      enter new name, or select one from above
   Tape mode . . . __            optional recording mode or density code









Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 169. Exported Stacked Volume Copy panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Exported Stacked Volume List panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager             Exported Stacked Volume List

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Type of listing SHORT         enter LONG to list logical volume details
   Logical vols(s) _____________________________________
                                 For LONG, enter logical volume(s)
                                 to be listed
                                 or
   Logical start . 0___          enter logical start volume
   Logical end . . 0____         enter logical end volume




Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 170. Exported Stacked Volume List panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Field Attributes panel - alphanumeric fields

This panel lists a field's formatting and data create attributes. You can modify the width of a field for TABL formatting, leading zero suppression for numeric fields. You can also specify data create attributes. The panel comes in two flavors, one for alphanumeric fields and one for numeric fields; this section describes the alphanumeric field version.

Panel and field definitions

```
Process   Options   Help
------------------------------------------------------------------------------
File Manager                  Field Attributes

Fixed Attributes:
          Field name  CONTRACTOR
                Type  AN
               Start  7
              Length  40
Use Attributes:
             Heading  ____________________
        Output width  ______
               CCSID  _____

Create Attributes:
              Filler  _______
              Action  __
     Start character  _
             Pattern  ____________________________________________________
 Repeat user pattern  ___           YES or NO
Scrambling Options:
 Scramble Type        Enter "/" to select option
 _ 1. Random          _  Value     Column In  ____________   Out  ____________
   2. Repeatable Dsn  ____________________________________________________
   3. Translate
   4. Exit
Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap      F10=Actions  F12=Cancel
```

Figure 171. Field Attributes panel - alphanumeric fields

When the selected field is alphanumeric, the fields on the Field Attributes panel are:

Fixed attributes
Shows the fixed attributes of the field, such as field name, type, and length, and the starting position, relative to zero, of the field within the record.

These attributes are determined by copybook definitions or by the user-defined attributes in a dynamic template.

Use attributes
Used to specify certain display (or print) attributes used by edit, view, and print functions. For alphanumeric fields, the available use attributes are:

Heading
The alternate heading that replaces the field name on displays and reports.

Output width
Number of character positions used by edit, view, compare and print functions to show this field in TABL display or print format. For alphanumeric fields, this value can range from 6 to either 30 or the field width + 10 (whichever is the greater).

The default output width is the maximum of the number of characters needed to show the field heading (or field name, if no heading is specified), and the number of characters needed to show the value of the field.

CCSID
Specifies the CCSID to be associated with this field.

Length Field
Only shows for a segmented template and alphanumeric fields whose length does not exceed 4.

Each layout can have one field flagged as a length field. This tells File Manager to calculate the segment length based upon the value contained in the field.

You can enter one of the following values:

blank This is not a length field.

1 The field value is the segment length.

2 The field value, plus the length of the field, is the segment length.

If a length field is not provided, the segment length is taken to be the 01 field length.

Create attributes
These define the field-specific attributes used by the data create functions when creating test data for records containing this field.

Filler Specifies a value that is to be placed in each byte of the field before any other operation in the construction of the field. Can be one of:

char A character such as `0` to be written in each byte.

X'cc' A hexadecimal value, such as `X'FF'` to be written in each byte.

Default value is blank.

Action
Specifies how the contents of the field are to be varied for each record. Can be one of:

FX The contents of the field are to remain fixed.

RA The contents of the field are to be generated from characters selected at random from the specified pattern. A different set of characters is selected for each record.

RO The contents of the field are to be rolled. The pattern you specify is shifted left one byte for each record until the first non-blank character of the pattern is in the first byte of the field. Then, the pattern is shifted right one byte for each output record until it returns to its original position in the field. RO can only be used with a user-supplied pattern, not with an IBM-supplied pattern.

The pattern must start with a blank, or the result is the same as FX. The roll only occurs within the length of the pattern.

RP The contents of the field are to be rippled. The pattern you specify is shifted left one byte for each record and the truncated character is appended to the right hand end of the pattern.

SL The contents of the field are to be shifted left for each record. The pattern you specify is shifted left one character

and filled to the right with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

SR The contents of the field are to be shifted right for each record. The pattern you specify is shifted right one character and filled to the left with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

TL The contents of the field are to be truncated on the left for each record. The pattern you specify is left truncated (the leftmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then, the original pattern is restored and the process is repeated.

TR The contents of the field are to be truncated on the right for each record. The pattern you specify is right truncated (the rightmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then, the original pattern is restored and the process is repeated.

WV The contents of the field are to be waved. The pattern you specify is shifted left one byte for each record until the first non-blank character of the pattern is in the first byte of the field. Then, the original pattern is restored and the process repeated. WV can only be used with a user-supplied pattern, not with an IBM-supplied pattern.

The pattern must start with a blank, or the result is the same as FX. The roll only occurs within the length of the pattern.

Start character

Sets the starting character to be used when you specify an IBM-supplied pattern (AL, AN, or CO) or a user-supplied pattern (with the exception of RO, WV, and FX). The specified character must be one of the characters in the IBM-supplied pattern or user-supplied pattern.

Default: First character in IBM-supplied pattern or user-supplied pattern.

Pattern

Specifies the pattern to be used when generating data for this field. You can specify either an IBM-supplied pattern indicator or a user-supplied pattern. The user-supplied pattern can be either a character string or a hexadecimal string. A character string must be enclosed in quotes, while a hexadecimal string must be enclosed in quotes and be preceded by a X. The hexadecimal string must contain an even number of valid hexadecimal characters. The IBM-supplied patterns you can specify are:

AL Alphabetic - characters A-Z, inclusive.

AN Alphanumeric - characters A-Z, 0-9 inclusive.

AX Alphanumeric displayed in long hexadecimal.

CO Collating sequence - all printable characters in the range X'40' - X'F9'.

If you provide a user-supplied pattern that is longer than the field, for actions other than RA and RP the pattern is first truncated on the right to fit the receiving field before the specified action is performed. For the RA action, characters are randomly selected from the entire user-supplied pattern. For the RP action, the entire pattern is rippled for each record before it is truncated to fit the receiving field. If you provide a user-supplied pattern that is shorter than the field, you can specify that you want the pattern to be repeated to fill the field.

Default: None

Repeat user pattern
Specify YES if you have provided a user-supplied pattern that is shorter than this field and you want the pattern to be repeated as many times as necessary to fill the field. By default, File Manager uses the fill character to pad the receiving field when the user-supplied pattern is shorter than the field. This option has no effect when the RA action is specified.

Note: IBM-supplied patterns are always repeated in a field, as necessary.

Default: NO

Scrambling Options
These options control field scrambling used during a copy function to produce test data.

Scramble Type
Specify one of the following values:

Blank No scrambling is performed. Value or range specifications are saved but ignored for the associated function.

1 (Random)
Performs random scrambling. The same input value produces different output values on subsequent invocations.

2 (Repeatable)
Performs repeatable scrambling. The same input value produces the same output value on subsequent invocations.

3 (Translate)
Performs translation. The value data set is searched to find a matching input value. If a match is found, then the output value is taken from the output column of the matching record.

4 (Exit)
Invokes a scrambling user exit. File Manager displays a panel where you can specify the user exit name and user exit parameters and options.

Value Select this option to provide values for the output field. The associated fields (**Column In**, **Out** and **Dsn**), in conjunction with the **Scramble Type**, determine the output value.

Column In
Defines the start location of the input field value on the value data set, and is used when the translate process is run during a copy operation to match the input field with a value on the value data set. The length of the field is set to the length of the input field that is mapped to this field during the copy process.

Note: This value is only required when you select the scramble type, **Translate**.

(Column) Out
Defines the start location of the output field value on the value data set, and is used as follows during a copy operation:

- For scramble type, **Translate**: If an input field value is matched on the value data set, then the corresponding output value is used.
- For scramble types, **Random** or **Repeatable**: The input value is used to randomly or repeatably select an output value from the value data set.

The length of the field is the current field length as displayed on this panel.

Note: This value is required when you select the scramble type, **Translate**. If you select **Random** or **Repeatable**, and also select **Value**, then the start location defaults to 1 if a value data set name (**Dsn**) has been provided.

Dsn Defines the value data set. It can be any cataloged sequential, partitioned or VSAM data set containing data that is used to determine the output field value during a copy process.

When specifying the data set name, the following rules apply:

- In a generic data set name, use the percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.
- The TSO prefix (as defined for your user ID) is used as the high-level qualifier for any name that is not enclosed in quotes.
- For performance reasons, you should qualify the data set name as much as you can.
- If the data set is partitioned, then you can provide a member name in the form *dsn*(*member*). If you do not provide a member name, or you provide a generic member name, then a member selection list is displayed for you to select a member.

Parent panels

Child panels

Related tasks and examples

"Setting the Data Create Attributes in your template" on page 246

Field Attributes panel - numeric field

This panel lists a field's formatting and data create attributes. You can modify the width of a field for TABL formatting, leading zero suppression for numeric fields. You can also specify data create attributes. The panel comes in two flavors, one for alphanumeric fields and one for numeric fields; this section describes the numeric field version.

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Field Attributes

Fixed Attributes:
          Field name  SERIAL-NO
                Type  ZD
               Start  47
              Length  6
Use Attributes:
             Heading  ____________________
        Output width  _____
       Leading zeros  NO          YES or NO
Create Attributes:
         Start value  ____________________
           End value  ____________________
           Increment  ____________________
               Cycle  __________
Scrambling Options:
 Scramble Type        Values
  _ 1. Random          _ 1. Range         Min  ____________   Max  ____________
    2. Repeatable        2. Value   Column In  ____________   Out  ____________
    3. Translate  Dsn  ____________________________________________________
    4. Exit
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 172. Field Attributes panel - Numeric Field

When the selected field is numeric, some fields differ to those displayed when the selected field is alphanumeric. Where the fields are different, the following definitions apply:

Use attributes
Used to specify certain display (or print) attributes used by edit, view, and print functions. For numeric fields, the available use attributes are:

Heading
The alternate heading that replaces the field name on displays and reports.

Output width
Number of character positions used by edit, view, and print functions to show this field in TABL display or print format. For numeric fields, from 6 to 30 (including a sign character and decimal point).

The default output width is the maximum of the number of characters needed to show the field heading (or field name, if no heading is specified), and the number of characters needed to show the value of the field.

Leading zeros
Specifies whether leading zeros are to be shown when this field is displayed or printed in SNGL or TABL display or print format. Specify YES if you want leading zeros shown. Specify NO if you want leading zeros replaced with blanks.

Default: YES

Length Field
Only shows for a segmented template and alphanumeric fields whose length does not exceed 4.

Each layout can have one field flagged as a length field. This tells File Manager to calculate the segment length based upon the value contained in the field.

You can enter one of the following values:

blank This is not a length field.

1 The field value is the segment length.

2 The field value, plus the length of the field, is the segment length.

If a length field is not provided, the segment length is taken to be the 01 field length.

Create attributes
These define the field-specific attributes used by the data create functions when creating test data for records containing this field.

Note: The mantissa consists of an optional sign (+ or -) followed by 1 to 16 digits. The mantissa can also contain a decimal point. The exponent consists of the letter E, an optional sign (+ or 0), and 1 or 2 digits.

Start value
Specifies the initial value you want a field to contain, before being adjusted by the increment you specify.

The value must be a number that, when converted to the appropriate numeric data type, the field can hold, and, for a field with decimal places, must not specify more decimal integers than the number of decimal places in the field definition. If you specify a negative number, the sign is honored, even if the field is unsigned.

If the field is a floating-point field, the start value can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01).

Default: 0

End value
Specifies the maximum value (if the increment is a positive number) or minimum value (if the increment is a negative number) you want a field to contain.

The value must be a number that, when converted to the appropriate numeric data type, the field can hold, and, for a field

with decimal places, must not specify more decimal integers than the number of decimal places in the field definition. If you specify a negative number, the sign is honored, even if the field is unsigned.

If the field is a floating-point field, the end value can be specified as a floating-point number consisting of a mantissa and an exponent (such as -01.14579E01).

Default: The largest positive or negative number the field can contain.

Increment
Specifies a positive or negative number by which you want the value in the field adjusted for each record (or cycle of records).

The increment must itself be a number that, when converted to the appropriate numeric data type, the field can hold, and, for a field with decimal places, must not specify more decimal integers than the number of decimal places in the field definition.

If the field is a floating-point field, the increment can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01).

For the first record (or cycle of records), the field is set to the start value you specify. For each subsequent record (or cycle of records), the increment is added to the value in the field. This process continues as long as the calculated value in the field does not exceed the specified end value.

Default: 0

Cycle Specifies the number of output records that are to be generated before the increment value is applied to the field value. For example, if you specify a field start value of 100, an increment value of 10, and a cycle value of 3, the field in the first 3 records contains 100, 110 in the next 3 records, 120 in the next 3 records, and so on.

If the cycle is 0, the value in the field is always set to the start value.

Scrambling Options
These options control field scrambling used during copy, import, or export functions to produce test data.

Scramble Type
Specify one of the following values:

Blank No scrambling is performed. Value or range specifications are saved but ignored for the associated function.

1 (Random)
Performs random scrambling. The same input value produces different output values on subsequent invocations.

2 (Repeatable)
Performs repeatable scrambling. The same input value produces the same output value on subsequent invocations.

3 (Translate)
Performs translation. The value data set is searched to find a matching input value. If a match is found, then the output value is taken from the output column of the matching record.

4 (Exit)
Invokes a scrambling user exit. File Manager displays a panel where you can specify the user exit name and user exit parameters and options.

Note: For numeric fields, setting **Leading zeros** to YES causes leading zeros to be scrambled when random or repeatable scrambling (**Value** option blank) has been requested.

For example:
- With **Leading zeros** set to YES, 00123 may produce 56872 (zeros changed).
- With **Leading zeros** set to NO, 00123 may produce 00343 (zeros unchanged).

For repeatable scrambling, setting **Leading zeros** to YES guarantees a unique result.

Values
For numeric fields, you can provide these options to control the output for the selected scramble type:

Blank File Manager determines the output value by performing either random or repeatable scrambling on the input value as determined by the scramble type.

1 (Range)
A range of values. You must provide a minimum and a maximum value that is valid for the field. During the copy operation, the scramble value generated is a number in the range provided. The minimum value must be less than the maximum value. The values provided must be between -2GB and +2GB.

2 (Value)
You provide values for the output field. The associated fields (**Column In**, **Out**, and **Dsn**), in conjunction with the scramble type, determine the output value.

Min Minimum value. Must be less than the maximum value and greater than or equal to -2GB. When option **1** (Range) is selected, you must specify a value in this field.

Max Maximum value. Must be greater than the minimum value and less than or equal to 2GB. When option **1** (Range) is selected, you must specify a value in this field.

Column In
Defines the start location of the input field value on the value data set, and is used when the translate process is run during a copy operation to match the input field with a value on the value data set. The length of the field is set to the length of the input field that is mapped to this field during the copy process.

Note: This value is only required when you select the scramble type, **Translate**.

(Column) Out
Defines the start location of the output field value on the value data set, and is used as follows during a copy operation:

- For scramble type, **Translate**: If an input field value is matched on the value data set, then the corresponding output value is used.
- For scramble types, **Random** or **Repeatable**: The input value is used to randomly or repeatably select an output value from the value data set.

The length of the field is the current field length as displayed on this panel.

Note: This value is required when you select the scramble type, **Translate**. If you select **Random** or **Repeatable**, and also select **Value**, then the start location defaults to 1 if a value data set name (**Dsn**) has been provided.

Dsn Defines the value data set. It can be any cataloged sequential, partitioned or VSAM data set containing data that is used to determine the output field value during a copy process.

If the **Translate** scramble type is selected, then the data set must contain the input and output values in the locations provided in the input and output columns.

If the **Random** or **Repeatable** scramble type is selected, then the data set must contain the output value in the locations provided in the output column.

Note: This field is required when you select the **Translate** scramble type. If you select **Random** or **Repeatable**, and also select **Value**, then you can optionally provide a data set name. If you leave this field blank having selected **Value**, you are prompted to enter the value list to be stored in the template.

When specifying the data set name, the following rules apply:

- In a generic data set name, use the percent sign (%) to represent a single character, and an asterisk (*) to represent any number of characters within one qualifier. Two asterisks (**) represent any number of characters within any number of qualifiers.
- The TSO prefix (as defined for your user ID) is used as the high-level qualifier for any name that is not enclosed in quotes.
- For performance reasons, you should qualify the data set name as much as you can.
- If the data set is partitioned, then you can provide a member name in the form *dsn*(*member*). If you do not provide a member name, or you provide a generic member name, then a member selection list is displayed for you to select a member.

Parent panels

Child panels

Related tasks and examples

"Setting the Data Create Attributes in your template" on page 246

Field Mapping panel

The Field Mapping panel describes the current relationship between receiving fields in the "To" (or "New") template or copybook to sending fields in the "From" (or "OLD") template or copybook. File Manager generates corresponding mappings based upon field names if you have issued the GM (Generate corresponding map) command prior to editing the mapping relationship; or the GE command (Generate and edit mapping).

This panel is displayed when "edit template" has been requested from the "To" panel of the Copy Utility, the Options panel of the Compare Utility or the Map To panel of the Template Workbench, and the "To" or "New" template or copybook contains only one record type, or when the E or S command is issued on the Record Type Mapping panel.

Note: If called from the Compare Utility, the panel displays all labels with "New" instead of "To" and "Old" instead of "From".

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                    Field Mapping                    Line 1 of 7

To           USERID.TEMPLATE(FMNCCPY2)
From         USERID.TEMPLATE(COPYTST1)

Cmd Lv To Field Name       Type  Len   Lv From Field Name             Type  Len
      ****  Top of data  ****
___  1 NEW-TYPE01           AN    84 |  1 REC-TYPE01                  AN    80
___  2 REC-TYPE             AN     2 |  2 REC-TYPE                   AN     2
___  2 NAME                 AN    20 |  2 NAME                        AN    20
___  2 SERIAL-NO            BI     4 |
___  2 AGE                  BI     2 |  2 AGE                         BI     2
___  2 SALARY               PD     4 |  2 SALARY                      PD     4
___  2 MONTH                BI     4 |  2 MONTH                       BI     4
      ****  End of data  ****

Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Describe
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 173. The Field Mapping panel

TO (or NEW) *template/copybook*
The name of the copybook or template that contains the record descriptions for the “To” or “New” data set.

FROM (or OLD) *template/copybook*
The name of the copybook or template that contains the record descriptions for the “From” or “Old” data set.

(To or New) Lev, Field Name, Type and Len
Shows the level number, field name, type and length of the fields in the “To” or “New” template.

(From or Old) Lev, Field Name, Type and Len
Shows the level number, field name, type and length of the fields in the “From” or “Old” template that have been mapped to the “To” or “New” field.

Cmd Prefix command area - used to enter a template editor prefix command. Enter `S` to change the field mapping or `E` to edit the new field attributes.

Parent panels

Child panels

Related tasks and examples

Field Selection/Edit panel

The Field Selection/Edit panel is primarily used to edit copybook templates. You can also edit some properties of dynamic templates within this panel, such as the record selection criteria, field sequencing and field selection. (You cannot modify the Field Name, Start, Length or Type fields; those fields can only be modified in the Dynamic Template panel.)

When you edit a template for segmented data, File Manager displays the Field Selection/Edit panel with an additional line identified by "**0 Rid**" showing the related ID expression (if any).

You can use the Field Selection/Edit panel (Figure 174 on page 566) to:

- Select which fields are displayed (when viewing or editing) or printed.
- Change the sequence in which fields are displayed or printed.
- Provide field headings that File Manager uses in place of the copybook or dynamic template field names, when displaying or printing.
- Begin editing field attributes
- Specify record identification criteria for the record type
- Specify record selection criteria for the record type
- Select fields to be included in a multi-segment key, and specify the order in which they are to be used in a keyed synchronization comparison.
- Display a subsequent panel from which you can choose which level-01 items to edit related ID criteria for an owning level-01.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 10

----------- Criteria - Enter 1 or 2 to specify expression by field -----------
1 Id : #2='02'                                                                +
2 Sel: ______________________________________________________________________ +
Offset ____0                                                          CCSID _____
Cmd Seq SHE Key Ref Field Name                          Picture  Type Start Length
                      ****  Top of data  ****
                  1 1 REC-TYPE01                                    AN      1     80
___ ___     ___   2  2 REC-TYPE                          XX        AN      1      2
___ ___      1    3  2 NAME                             X(20)     AN      3     20
___ ___     ___   4  2 EMPLOYEE-NO                      9(4)      BI     23      2
___ ___     ___   5  2 AGE                              9(4)      BI     25      2
___ ___     ___   6  2 SALARY                           9(7)      PD     27      4
___ ___     ___   7  2 MONTH OCCURS 12 TIMES            9(8)      BI     31      4
___ ___     ___   8  2 FILLER                           XX        AN     79      2
                    ****  End of data  ****

Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 174. Field Selection/Edit panel

```
 Process   Options   Help
 ------------------------------------------------------------------------------
File Manager                Field Selection/Edit                 Line 1 of 6

---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
0 Rid: SEG(HEADER-01):#5: TYPEREC    ='01'                                    +
1 Id : ______________________________________________________________________ +
2 Sel: ______________________________________________________________________ +
Offset ____0          Enter "/" to OR with related ID              CCSID _____
Cmd Seq SHE Ref  Field Name                            Picture  Type Start Length
                 ****  Top of data  ****
              1  1 DETAIL-0101                                    AN      1     30
___ ___       2   2 SUBTYPE-FIELD                                 AN      1      9
___ ___       3    3 SUBTYPECONST                    X(7)        AN      1      7
___ ___       4    3 TYPEREC                         X(2)        AN      8      2
___ ___       5   2 DETAILTEXT01                                 AN     10     21
___ ___       6    3 DETAIL01-01                     X(21)       AN     10     21
                 ****  End of data  ****


Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 175. Field Selection/Edit panel - alternative format for a template for segmented data

0 Rid Used to specify the related ID expression. This line only appears on the

Field Selection/Edit panel when you are editing a template for segmented data (which you indicate by selecting the **Template for segmented data** processing option on the previous panel (Record Type Selection).

1 Id Used to specify the Record identification criteria that File Manager should use to identify this record type. In the absence of record identification criteria, File Manager uses the record length to identify record type.

For fixed-length records (as defined by the record definition in the template, not the data set attributes), the length of the record read must exactly match the length of the corresponding record definition in the template. For variable-length records, the length of the record read must be equal to or greater than the minimum length and equal to or less than maximum length of the corresponding record definition in the template.

If File Manager cannot match the length of a record to one of the record definitions in the template, the record is not selected for processing. If the length of a record matches more than one record definition in the template, and no explicit record identification criteria is specified, File Manager uses the first matching record type in the template. Therefore, if the record length of each different record type in the file is not unique, you should use this field to specify some unique identification criteria.

Enter 1 to display the Record Identification Criteria panel and specify the criteria by field or type a free form REXX expression in the adjacent field. You can scroll to the right or left within the field, using the Right (F11) or Left (F10) function keys, or you can expand the field, using the Expand function key (F4).

Specifying criteria by field or by free form REXX expression is mutually exclusive. If you select option 1 and enter criteria by field, any free form REXX expression entered in the adjacent field is replaced by the new criteria. You cannot manually edit this expression. To restore the ability to create a freeform REXX expression for record identification, you would need to clear any criteria entered in the Record Identification Criteria panel.

2 Sel Used to specify the Record selection criteria that File Manager should use to select records for processing. You can use this field to limit the records of a given record type that you want to process. If you do not provide any record selection criteria, all records of this record type are selected for processing (if the record type itself has been selected for processing).

Enter 2 to display the Record Selection Criteria panel and specify the criteria by field or type a free form REXX expression in the adjacent field. You can scroll to the right or left within the field, using the Right (F11) or Left (F10) function keys, or you can expand the field, using the Expand function key (F4).

Specifying criteria by field or by free form expression is mutually exclusive. If you select option 2 and enter criteria by field, any free form REXX expression entered in the adjacent field is replaced by the new criteria. You cannot manually edit this expression. To restore the ability to create a freeform REXX expression for record selection, you would need to clear any criteria entered in the Record Selection Criteria panel.

Offset Offset field - used to enter an offset value. An offset value is a negative or positive integer, between -32760 and +32760, that is added to the record length of the Level 01 field and to the starting position of all fields within

the record type, shifting the layout left or right in relation to the records being processed. An offset value of 0 removes a previously supplied offset.

Enter "/" to OR with related ID
By default, identification criteria are ANDed with related ID criteria if both are specified. Select this option to OR the identification criteria with any specified related ID criteria.

CCSID
The CCSID to be associated with all alphanumeric fields in the record unless the field has a CCSID associated with it. If this field is changed those alphanumeric fields in the record which have the same CCSID will have the CCSID associated with it changed.

Cmd Prefix command area - used to enter a template editor prefix command.

Seq Sequence field - used to order fields for display in File Manager editor sessions and printing data via the File Manager Print utility. The default display/print order is "selected fields in field reference order".

SHE Select/Hold/Edit field - shows the field status:

- **S** The field has been selected for processing.
- **H** The field has been selected to be held on the edit display.
- **E** The field has been previously edited to add information such as data create, field headings, and scrambling options.

Ref Field reference - shows the field reference number assigned by File Manager to the field name. Field reference numbers are assigned to all fields in a record, and are used to identify fields in identification or selection criteria expressions.

Key Key segment sequence. Shows the key segment sequence numbers used to define a multi-segment key field for data set comparison. The display of the Key column can be toggled on or off using the KEYFLDS command.

Field Name
Shows the level number and field name of the field, together with other attributes such as redefines, array sizes and depending on references. If the field is an array, the dimensions are shown in brackets after the field name. If the field has been selected for processing, the field name is highlighted.

Picture
Shows the picture clause for COBOL and for PL/I. Shows the length and scale (if non zero) for binary and packed fields. Also shows the bit length for bit fields.

Type Shows the data type of the field:

- **AN** Alphanumeric (includes alphabetic, and group items)
- **AX** Alphanumeric displayed in long hexadecimal.
- **BI** Binary
- **BT** Bit
- **DB** DBCS
- **FE** Floating point (external)
- **FP** Floating point (internal)
- **G** Graphic

PD Packed decimal (internal decimal)

VC Variable Character

VD Variable DBCS

VG Variable Graphic

ZC Character Null terminated

ZD Zoned decimal (external decimal)

ZE Zoned decimal edited

ZG Graphic Null terminated

Z2 DBCS Null terminated

Numeric data types supported by File Manager include numeric-edited data types.

Start Shows the start column of the field. For variable located fields the start location is based on the maximum length of the record.

Length
Shows the length of the field. For a level-01 field, if the record is variable length, the length shown is the maximum record length.

Available commands

The prefix commands that can be entered in the Cmd field are:

E Displays the Field Attributes panel for this field. You can use the Field Attributes panel to specify the following:

- For a numeric field, whether you want leading zero suppression when the field is displayed on the screen or printed using the SNGL or TABL display or print formats.
- When you use the template to help generate test data, what data pattern File Manager should use for the field.

H Set Hold status for a single field, if the field is already in Hold status the Hold status is removed. When a field has the Hold status set it remains in the editor window during left/right scroll operations.

Hn Set Hold status for n fields, if any of the fields are already in Hold status, the Hold status is removed.

HH Select a block of fields to have the Hold status set. If any of the fields are already in Hold status, the Hold status is removed. Use the HH prefix command to mark the start and end of the block of fields.

S Select a single field for processing or, if the field is already selected for processing, deselect it. If you select multiple fields for processing using the S prefix command, the fields are displayed in the order they appear in the template.

Sn Select *n* fields for processing or, if any of the fields are already selected for processing, deselect them.

S* Select all fields from the current field to the end of the list for processing or, if any of the fields are already selected for processing, deselect them. If you want to exclude just a few fields, you can use S* to explicitly select all the fields in a fresh template, then use S to deselect the fields you want to exclude.

SS Select a block of fields for processing or, if any of the fields are already selected for processing, deselect them. Use the SS prefix command to mark the start and end of the block of fields.

X Toggle between a display type of AN (character) and AX (long hexadecimal). This command has no effect on non-alphanumeric fields.

X*n* Perform the X command against all fields from the current field for *n* fields.

X* Perform the X command against all fields from the current field to the end of the list.

XX Perform the X command against a block of fields. Use the XX prefix command to mark the start and end of the block of fields.

The primary commands that are available on this panel are:

- "DESCRIBE primary command" on page 760
- "FIND/FX primary command" on page 769
- "RESET primary command" on page 806
- "RUNTEMP primary command" on page 812
- "SAVE primary command" on page 812
- "SAVEAS primary command (templates)" on page 813
- "SELECT primary command" on page 814
- "TPRINT primary command" on page 825

Parent panels

Child panels

Related tasks and examples

Field Selection List panel

The Field Selection List panel provides a list of fields in the current template. This panel is entered if no field is identified when the FMAP command is entered.

Panel and field definitions

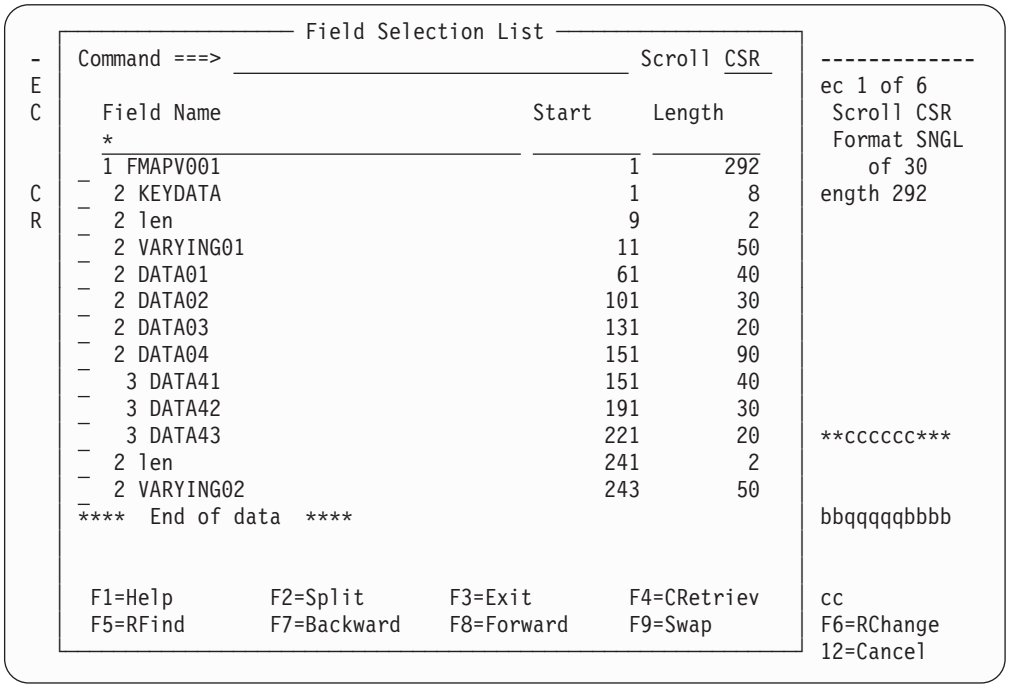


Figure 176. Field Selection List panel

Field Name
The fields in the current template.

Start The start column of the field.

Length
The length of the field.

Parent panels

- <xref href="editpan.dita#editpan"/>
- <xref href="viewpan.dita#viewpan"/>

File Selection panel

The File Selection panel provides a list of HFS directory entries, from which you can select the files to be included in your current process. This panel is displayed when you have specified a HFS directory on an Entry panel.

Note: The title of the File Selection panel shown in Figure 177 on page 572 includes the word, “View”, indicating that it was invoked from the View Entry panel. The title varies according to which Entry panel invokes the File Selection panel.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
  File Manager              View File Selection
 PATH   /u/acook
   Name             Prompt   Typ  Size     Created  Changed           ID
   *                *        *    *        *        *                 *
   .                         Dir     8192  09/05/01 09/05/01 10:21:05 IBMUSER
   ..                        Dir     8192  09/05/01 09/05/01 10:15:46 IBMUSER
   .profile                  File      55  09/05/01 04/10/13 09:40:45 IBMUSER
   .sh_history               File     941  09/05/01 07/05/02 11:29:02 IBMUSER
   acook.settings            File     677  09/05/01 04/10/13 09:52:09 IBMUSER
   sdsds                     File       0  09/05/01 06/08/23 10:32:48 IBMUSER
 ****  End of data  ****




 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
  F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 177. File Selection panel

(Selection field)
Selection field. Entering an "S" results in one of the following:

- A single file is selected and returned to the calling routine. (Subsequent selections are ignored.)
- Toggling of the selected value in the Prompt field (Size column is used).
- Selects a file to be processed by the function that invoked the list.
- Displays the list of entries on the next level (when directory selected).

Name An HFS object name. If the full name is not displayed, position the cursor at the name and press Enter to display a pop-up window showing the full name.

If the Copy From panel invoked the File Selection panel, the pop-up window also allows you to change the target name.

Prompt
A dynamic area that displays the last action that occurred for the listed file. It can have the following values:

Browsed
File was browsed

Edited File was edited

Printed
File was printed

Nocopy
File was not copied

Copied
File was copied

Repl File (member) was replaced

Norepl
File (member) was not replaced

Error Error occurred when processing file

Invname
Copy to a PDS library: file name cannot be transformed to a member name

Invtype
HFS object not serviced by File Manager

Notauth
You are not authorized to access the file

Invrecl
Record size is greater than the allowable maximum (text mode)

Typ The HFS object type:

Dir Directory

Char Character special file

File Standard HFS file

FIFO Pipe or FIFO special file

Syml Symbolic link

Sckt Socket file

Size The object size in bytes. For large files, kilobytes (K), megabytes (M), or gigabytes (G) are used.

Created
The date the HFS object was created in YY/MM/DD format.

Changed
The date and time the HFS object was last changed in YY/MM/DD-HH:mm:SS format.

ID Owner ID.

Parent panels

- "Browse Entry panel" on page 459
- "Edit Entry panel" on page 541
- "Copy From panel" on page 491
- "Copy To panel" on page 497
- "Print Utility panel" on page 620
- "Compare Utility: "Old" and "New" panels" on page 471

Child panels

The display panel that would normally result from the initiating Entry panel.

Related tasks and examples

- Chapter 11, "Using UNIX System Services and the Hierarchical File System," on page 391
- "Manipulating your view of selection lists" on page 28

Find/Change Utility panel

The Find/Change Utility allows you to search for or change a string in a PDS, a VSAM data set, or a sequential data set by entering a FIND or CHANGE command on the Command line. You can also search for strings in HFS files.

Panel and field definitions

```
 Process   Options   Help
-------------------------------------------------------------------------------
File Manager                Find/Change Utility

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name  'FMNUSER.DATA'                                  +
   Member . . . . . *          (Blank - selection, pattern - process list)
   Volume serial  . _____      (If not cataloged)
   Record count . . ALL        (Number of records to be searched)
- Additional options
Listing data set  . SRCHFOR.LIST

Enter "/" to select option                        Listing Option  ISPF Packing
_ JCL Source format          _ Immediate change    1 1. Long       1 1. Asis
_ Use REXX proc ________     _ Batch execution       2. Summary      2. Pack
_ REXX no update             _ Directory integrity Stats Option      3. Unpack
_ Advanced member selection                        2 1. Off          4. None
  Binary mode, reclen        _ CAPS initially on     2. Force        5. Skip
Process List:
Sel Name     Prompt   Alias-of Size Created  Changed           ID
_   M100     Selected
_   M1000    Selected
_   M10000   Selected
_   M10001   Selected
_   M10003   Selected
_   M10005   Selected
_   M10006   Selected
_   M10007   Selected
_   M10008   Selected
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=Refresh   F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 178. Find/Change Utility panel

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key to display a pop-up window in which you can enter a longer name.

Member
If you specified the name of a partitioned data set (PDS) without including

a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set. Required for data sets which are not cataloged.

Record count
Number of logical records to be searched for FIND/CHANGE/FINDNOT commands. For a PDS, the number of logical records to be searched per member.

Range = 1 - 99,999,999; default = ALL.

Additional options
Collapses or expands the following section of the Find/Change Utility panel.

When a minus sign (“-”) is shown, position the cursor on the minus sign and press Enter to expand the following section of the panel.

When a plus sign (“+”) is shown, position the cursor on the plus sign and press Enter to collaps the following section of the panel.

Listing data set
Specifies the data set where File Manager find/change results are to be stored. Use the default name or enter a sequential data set name.

Default: '*userid*.SRCHFOR.LIST'

JCL source format
Indicates that the data set contains JCL and that the JCL syntax is to be preserved.

If not successful at maintaining the number and size of records, File Manager attempts to rewrite the file:

- More errors are possible in this case. For example, a PDS(E) may run out of room.
- If a logical line is changed and requires more physical records, the file is rewritten. The data in columns 73–80 for new physical records is copied from the last related original physical record.

The file must be non-VSAM and have a fixed record length of 80.

When using the **JCL source format** option, the columns searched are set to 3 through 71, unless the statement is not a JCL statement. A statement is considered to be a JCL statement if it begins with the strings "/*" or "//". If the statement does not begin with either of these strings, it is not considered to be a JCL statement in which case any column range specified on the FIND (or CHANGE, respectively) command or preset using the BOUNDS command is honored. If no column range has been specified, the full record is searched.

Use REXX proc
You can use this option to perform either of these actions

- Enter a temporary REXX procedure for one-time use by entering a single asterisk (*). File Manager displays an Edit panel, in which you can create a new REXX procedure.

- Specify the name of the member containing the REXX procedure you want to use, The member must belong to the PDS allocated to ddname FMNEXEC. You can enter any of the following:
  - The name of the member.
  - A member name pattern (other than a single *) to list all matching members. You can then select the required member by entering an `S` in the **Sel** field. A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

    asterisk (*)
    Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter `*d*`, a list of all members in the data set whose name contains " d" is displayed.

    percent sign (%)
    A place-holding character representing a single character. As many percent symbols as necessary can appear anywhere in a member name. For example, if you enter `%%%%`, a list of all members in the data set whose name is four characters in length is displayed.

Note: If you select this option but leave the **Use REXX proc** member entry field blank, File Manager displays a member name list. You can then select the required member by entering `S` in the **Sel** field.

(Also, see "Supplying a procedure when using a File Manager panel" on page 417.)

REXX no update
Allows you to specify that you intend no updates to the FCH data set while executing the utility. This option is valid only when a REXX procedure has been specified and is ignored otherwise. If selected, it forces the allocation of the data set as input only. All updates to the data are ignored.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The field only displays if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. An I/O exit can only be used to process the data set in which you are creating records. It cannot be used to process the copybook or template that you are using to format the data set.

Immediate change
When you use the CHANGE command, the input data set is updated immediately (without displaying the changes in the listing data set).

Batch execution
Creates JCL to reflect the command entered. The JCL is presented in an Edit session which you can edit before submitting.

Batch execution restricts the member selection to the pattern specified in the member field. Batch execution does not produce a pop-up selection panel for member selection. If you leave the member field blank, an asterisk (*) is substituted in the JCL generated. For more information, see "FCH (Find/Change)" on page 1015.

Stats Determines whether ISPF statistics (if present) for the PDS members being processed are updated:
Blank Update ISPF statistics.
1 Off Do not update ISPF statistics.
2 Force
Always update or create ISPF statistics.

Directory integrity
Forces an override of the default PDS(E) member processing method which allows for faster PDS directory access.

This option has significant performance impact. When selected, the members are processed in a way which allows concurrent directory updates as File Manager accesses the members using current directory information.

When not selected, the member processing is performed faster, but may be affected by PDS(E) directory updates, possibly causing I/O errors if the data set is updated concurrently.

Listing Option
Determines the format of the output report.

1 A full report, including each record found or changed.

2 A summary report providing totals for records processed and strings found and changed.

ISPF Packing
Provided that the data set is a sequential, PDS or PDSE file and an I/O exit routine is not used, one of these options can be used to control the utility's behavior when processing data that is in ISPF PACK format.

1. Asis
If the data set is packed, it is unpacked before any processing. The data set is rewritten in packed format only when it was packed initially.

2. Unpack
If the data set is packed, it is unpacked before processing. The data set is always rewritten in unpacked format.

3. Pack
If the data set is packed, it is unpacked before processing. The data set is always rewritten in packed format.

4. None
No checking or processing of ISPF packed data occurs. The FIND

and CHANGE commands operate on the packed data. This option is forced if an I/O exit has been used.

5. Skip
If the data is packed, no processing occurs.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the Binary mode option is selected, records are derived based on the fixed record length specified. The default is 80. Can be in the range: 1–32760.

Available commands

- "BOUNDS primary command" on page 743
- "CAPS primary command" on page 744
- "CHANGE/CX primary command" on page 748
- "FIND/FX primary command" on page 769
- "FINDNOT primary command" on page 777
- "LOCATE primary command" on page 786
- "REFRESH primary command" on page 804
- "RESET primary command" on page 806
- "SELECT primary command" on page 814
- "SORT primary command" on page 818
- "VCONTEXT primary command" on page 827

Parent panels

- "Utility Functions menu panel" on page 703

Equivalent functions

- "FCH (Find/Change)" on page 1015

Related tasks and examples

- "Finding and changing data in multiple PDS members" on page 270
- "Selecting a range of PDS(E) members" on page 34

FMAP Copybook or Template panel

<!-- PM80797 -->

This panel is entered if no template information is provided with the FMAP command.

Panel and field definitions

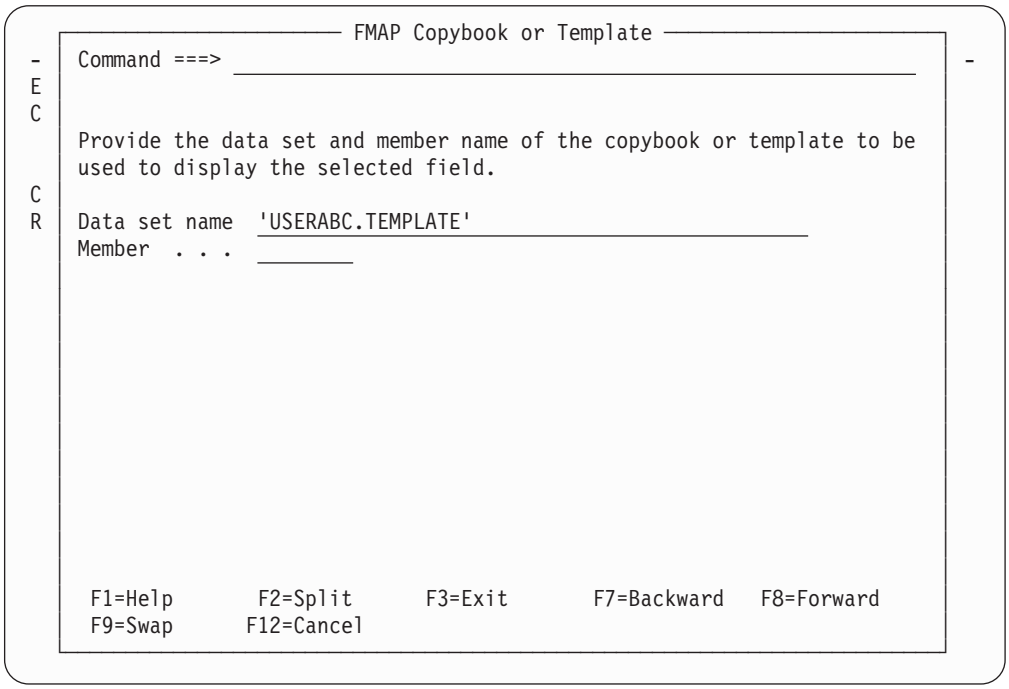


Figure 179. FMAP Copybook or Template panel

Data set name
Data set name of the template or copybook to be used to format your data.

Member
Member name of the template or copybook to be used to format your data.

From (or Old) Field Mapping panel

The From Field Mapping panel (or Old Field Mapping panel) provides a list of fields in the "From" or "Old" copybook or template that can be mapped to the selected field in the "To" copybook or template. If called from within the Copy Utility or the Template Workbench and the field does not qualify as a sending field for the selected TO field, the prefix command area is protected (see "Field mapping rules" on page 192).

This panel is displayed when the E prefix command is entered against a field on the Field Mapping panel.

Note: If called from the Compare Data Utility, the panel displays all labels with "New" instead of "To" and "Old" instead of "From".

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 From Field Mapping
To   template/copybook : FMNUSER.TEMPLATE(FMNCCPY1)
From template/copybook : FMN.SFMNSAM1(FMNCCPY)
To field . . . . . . . : #4 EMPLOYEE-NO
From field . . . . . . : #5 EMPLOYEE-NO
Corresponding(Y/N) . . : N               (Auto map for group items).

Sel   Ref Lvl From Field                                        Typ Start Length
              ****  Top of data  ****
___     D --  Delete "From field"
___     1  1  REC-TYPE01                                        AN      1     80
___     2  2  REC-TYPE                                          AN      1      2
___     3  2  REC-ID                                            AN      1      2
___     4  2  NAME                                              AN      3     20
___     5  2  EMPLOYEE-NO                                       BI     23      2
___     6  2  AGE                                              BI     25      2
___     7  2  SALARY                                           PD     27      4
___     8  2  MONTH(12)                                        BI     31      4
___     9  2  FILLER                                           AN     79      2
Command ===>                                                      Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F12=Cancel
```

Figure 180. The From Field Mapping panel

To (or New) template/copybook
The name of the copybook or template that contains the record descriptions for the "New" data set.

From (or Old) template/copybook
The name of the copybook or template that contains the record descriptions for the "Old" data set.

To (or New) field
Field reference number and name of selected "New" field

From (or Old) field
Field reference number and name of selected "Old" field

Corresponding
Used to specify whether File Manager should automatically map subordinate elementary fields when the “To” or “New” and “From” or “Old” fields are group fields. Specify Y if you want each elementary field in the “From” or “Old” group field mapped to a field in the “To” or “New” group field with the same name. Specify N if you only want the “From” or “Old” field mapped to the “To” or “New” field as though both fields were elementary fields.

Sel Select field - used to select the “From” or “Old” field you want to map to the “To” or “New” field. You can select the field by entering S in this field.

Ref From field reference - shows the field reference number assigned by File Manager to the field name in the “From” or “Old” template.

Lev and Field Name
Shows the level number and field name of fields in the “From” or “Old” template that can be validly mapped to the “To” or “New” field. If the selected “To” or “New” field is a level-01 field, only level-01 fields in the “From” or “Old” template are listed. For a “To” or “New” field other than

a level-01 field, only those “From” or “Old” fields that can validly be moved to the selected “To” or “New” field are listed for selection.

Type Shows the data type of the “From” or “Old” field.

Start Shows the start column of the “From” or “Old” field.

Length
Shows the length of the “From” or “Old” field. For a level-01 field, if the record is variable length, the length shown is the maximum record length.

Delete "from field" (or Delete "old field")
Select this entry if you want to clear the “From” or “Old” field that is currently selected for mapping to the “To” or “New” field.

Available commands

- “FIND/FX primary command” on page 769
- “CHANGE/CX primary command” on page 748
- “BOUNDS primary command” on page 743
- “CAPS primary command” on page 744
- “LOCATE primary command” on page 786

Parent panels

Child panels

Related tasks and examples

GDG Entry Detail panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  GDG Entry Detail                   Line 1 of 6

GDG Catalog Entry:
   Data set name . 'SYATES.UNITTEST.RFM0189.GDG1'
   Catalog ID  . . 'CATALOG.UCATAPC'

GDG BASE Attributes:
   Creation date . 2003.351           Expiration date (NONE)
   Limit . . . . . 255      Maximum number of generations
   Scratch . . . . N                  Empty . . . . . N

GDG Associations:
   Related NONVSAM 'SYATES.UNITTEST.RFM0189.GDG1.G0001V00'
                   'SYATES.UNITTEST.RFM0189.GDG1.G0002V00'
                   'SYATES.UNITTEST.RFM0189.GDG1.G0003V00'
                   'SYATES.UNITTEST.RFM0189.GDG1.G0004V00'
                   'SYATES.UNITTEST.RFM0189.GDG1.G0005V00'
                   'SYATES.UNITTEST.RFM0189.GDG1.G0006V00'
                   ****  End of data  ****
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 181. GDG Entry Detail panel

Parent panels

- "Catalog Services Data Set List panel" on page 466
- "Display VTOC Data Set List panel" on page 528

Child panels

Related tasks and examples

IAM KSDS Define panel

You can use the IAM ESDS Define or IAM KSDS Define panel to specify the allocation attributes for IAM data sets. There are two versions of this panel, one for each supported IAM data set type (ESDS and KSDS).

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  IAM KSDS Define

NONVSAM Catalog Entry:
   Data set name  . . 'USERID.NEWDS3'
   Catalog ID . . . . _____________________________________________________
                                                                    More:     +
VSAM Associations:
   VSAM data type . . KSDS       Expiration date  . ________
   Dataset Owner id . $IAM____

IAM Override information (if any)

--------------------------------------------------------------------------------
VSAM Cluster Attributes:
   Key length . . . . _____      Key offset . . . . _____
   CI size  . . . . . _____      size of the data control intervals
   Buffer space . . . ________   buffer space to be allocated at open time
   Shr cross region . _
VSAM Data Allocation:
Press ENTER to define the catalog entry or EXIT to cancel
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 182. IAM KSDS Define panel

NONVSAM Catalog Entry - Data set name
The name of the catalog entry, from one to 44 characters. For DEFINE and ALTER commands, enter the new name.

NONVSAM Catalog Entry - Catalog ID
Name of a catalog that contains the entry. Default: System catalog search order.

Dataset Owner id
Specify the ownerid of the data set. Note that IAM files are required to have either an OWNER of $IAM, or the word $IAM somewhere within the data set name. If this is not the case, the define process may create a VSAM cluster instead of a IAM data set.

IAM Override information
This allows you to enter any parameters for the CREATE override

statement when defining a IAM file. This is an optional field. Any information entered here is appended to a CREATE statement generated internally by File Manager.

VSAM Associations, Cluster Attributes, and so on
For general information about defining data sets, see the *DFSMS Using Data Sets* manual appropriate for your operating system. For specific information about valid field values, use File Manager's Field-level Help (put your cursor in the field and press F1).

Parent panels

"Allocate panel" on page 448

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

"Allocating a new data set" on page 246

IAM Entry Details panel

The Catalog Services: IAM Entry Detail panel is used to display the parameters of a selected IAM data set, allowing you to review or modify them.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 1 of 32

IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:
IAM100 IAM FILE ANALYSIS - DSN=FMNUSER.TESTMV.IAM.ESDS
       ------------------------------------------------------------
       FILE FORMAT -- =  ENHANCED  -  FILE STATUS ----------- =  LOADED
       RECORD SIZE -- =     27990  -  FREESPACE - CI% ------- =       0
       CI SIZE ------ =     32768  -  FREESPACE - CA% ------- =       0
       BLOCK SIZE --- =     32760     EXTENDED PE ----------- =    8635 BLOCK
       BLOCK FACTOR - =         1  -  REQUESTED OVERFLOW ---- =         0 RECS

Command ===> ____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up         F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 183. Example: Catalog Services parameters for an IAM entry (first panel)

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 8 of 32
IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:

       VAR OVERFLOW - =        NO  -  EXTENDED OVERFLOW ----- =         0 BLOCK
       FILE TYPE ---- =      ESDS  -  EXTENDED ALLOCATED ---- =      8641 BLOCK
       DEVICE TYPE -- =      3390  -  EXTENDED AVAILABLE ---- =         0 BLOCK
       VOLUME COUNT - =         5  -  SPACE USED ------------ =      8850 TRACK
       VOLSER ------- =    MVS4W5  -  SPACE ALLOCATED ------- =      3240 TRACK
       VOLSER ------- =    MVS4W4  -  SPACE ALLOCATED ------- =      1635 TRACK
       VOLSER ------- =    MVS4WA  -  SPACE ALLOCATED ------- =      1965 TRACK

Command ===> _____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 184. Example: Catalog Services parameters for an IAM entry (continued)

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 15 of 32
IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:

       VOLSER ------- =    MVS4WB  -  SPACE ALLOCATED ------- =      1005 TRACK
       VOLSER ------- =    MVS4W8  -  SPACE ALLOCATED ------- =      1005 TRACK
       TOTAL EXTENTS  =        55  -  TOTAL SPACE ALLOCATED - =      8850 TRACK
       PRIMARY SPACE  =        67  -  SECONDARY SPACE ------- =        67 CYL
       MULTIVOLUME -- = SECONDARY  -  MAX SECONDARY --------- =        67 CYL
       RELEASE ------ =        NO  -  SHARE OPTIONS --------- =         2
       DATA COMPRESS  =   ENABLED  -  INDEX COMPRESS -------- =        NO

Command ===> _____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 185. Example: Catalog Services parameters for an IAM entry (continued)

```
 Process   Options   Help
-------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 22 of 32
IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:

       TOTAL RECORDS  =    110173  -  INSERTS --------------- =     102173
       UPDATES ------ =         0  -  DELETES --------------- =          0
       HIGH USED RBA  =3610148864  -  HIGH ALLOCATED RBA ---- =  289926000
       FILE DEFINED   =  2001.331  -  11/27/2001 -  2:07 PM - =   14:07:39
       FILE LOADED -- =  2001.331  -  11/27/2001 -  2:07 PM - =   14:07:41
       LAST UPDATED - =  2001.331  -  11/27/2001 -  2:12 PM - =   14:12:33
       STORAGE REQUIRED FOR PRIME INDEX ------------- =       824

Command ===> ____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up         F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 186. Example: Catalog Services parameters for an IAM entry (continued)

```
 Process   Options   Help
-------------------------------------------------------------------------------
File Manager                  IAM Entry Detail                  Line 29 of 32
IAM NONVSAM Entry details:
   Catalog Id . . . . 'CATALOG.UCATAPC'
   Data set name  . . 'FMNUSER.TESTMV.IAM.ESDS'
   VSAM data type . . ESDS          (KSDS or ESDS)
   Creation date  . . 2001.331      Expiration date  . (NONE)
   SMS managed  . . . Y             Data class . . . . *UNKNOWN
   Storage class  . . BASE          Management class . STANDARD
   Last backup date . 0000.000.0000

IAM details from IAMPRINT:

       NUMBER OF IAM DATA BLOCKS ------------------- =        208
       EXTENDED HIGH ALLOCATED RBN ----------------- =       8849
       ------------------------------------------------------------

****  End of data ****

Command ===> ____________________________________________________ Scroll CSR
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up         F8=Down
F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 187. Example: Catalog Services parameters for an IAM entry (continued)

For more information about the fields in this panel, refer to your IAM documentation or press F1 in any field.

Parent panels

- "Catalog Services Data Set List panel" on page 466

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "IAM Entry Detail display" on page 318

Initialize Tape panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Initialize Tape

Tapes:  No tapes allocated


Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Volume serial . ______       volume identifier or blank for NL tape
   Data set name . ____________________________________________
   ASCII format  . NO           YES or NO
   Scale percent . 0            optional scale percentage or 0 for all









Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 188. Initialize Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Key positioning panel

Panel and field definitions

```
 Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────────
          Key Positioning                                        Line 1 of 2

Data set: FMN.RFM0740.HOGN.KSDS
Field          Data
M-KEY                          0









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 189. Key positioning panel

Formatted key positioning allows you to change the current key location using the template provided. All field values that are required to map the key area are displayed using SNGL format.

To control the appearance of the SNGL display, these SNGL display primary commands are available:

HEX
PIC
RDF
REF
SLOC
STR

To scroll the key display, use these commands:

BOTtom
DOWN
TOP
UP

To search the display area, use these commands:

Find
Locate

01 Layout Selection List panel

Panel and field definitions

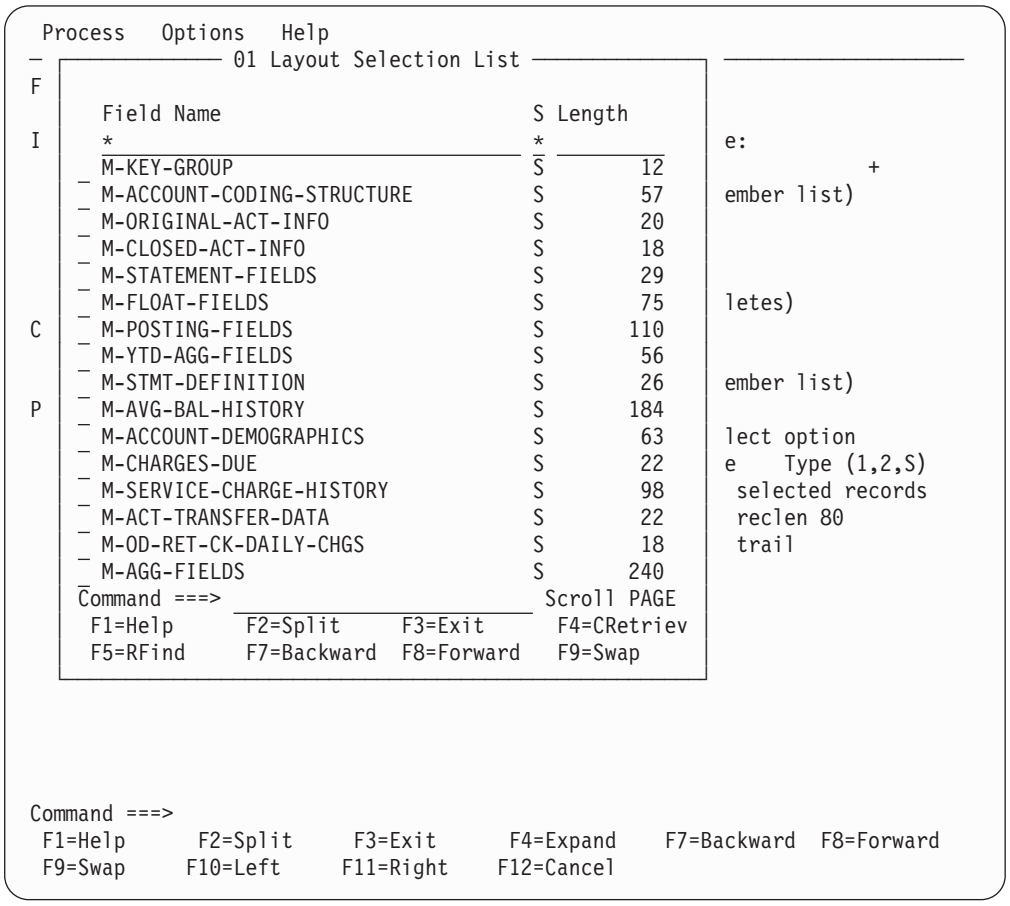


Figure 190. 01 Layout Selection List panel

This panel is entered to select a layout to be used by the calling command. Use the prefix command S or / to select the layout to be used.

The columns displayed are as follows:

Field Name
The 01 field names in the current template.

S Whether the layout is currently selected for processing.

Length
The length of the associated layout.

Library List panel

The Library List panel is used to specify the data sets in which multiple copybook members can be found. Up to 12 data set libraries can be specified.

Panel and field definitions

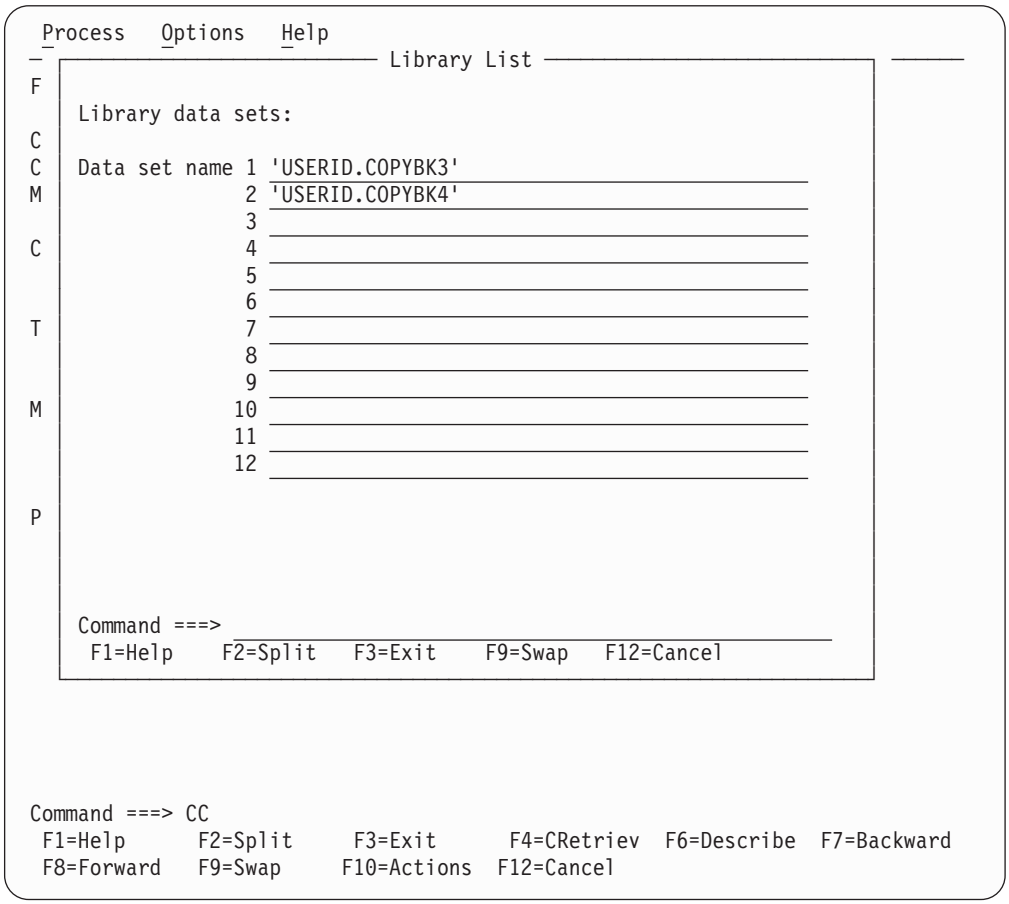


Figure 191. Library List panel

Data set name

The name of the SYSLIB data sets in which your copybooks reside. The first entry is populated with the copybook name specified on the Template Workbench panel. Any libraries previously identified are also displayed. Up to 11 additional SYSLIB data sets can be specified (total of 12 SYSLIB data sets). Up to 12 data set libraries can be specified. The data sets may be PDSs, PDSEs, CA-Panvalet libraries, or other library management system libraries. You can specify multiple PDSs, PDSEs, CA-Panvalet and other library management system libraries but they must be either all PDSs or PDSEs, or all CA-Panvalet, or all the same library management system libraries. You cannot mix library types.

All copybooks to be included in a template must be of the same language, that is, all COBOL or all PL/1.

Library members may not be packed by ISPF.

Parent panels

- "Template Workbench panel" on page 698

Child panels

- "Copybook Selection panel" on page 505

Related tasks and examples

Load Module Compare - entry panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                Load Module Compare

"Old" Partitioned Data Set:
   Data set name . . . . . 'FMN.RFM0411.COPY'
   Member  . . . . . . . . RFM0411      Blank or pattern for member list
   Volume serial . . . . . ______       If not cataloged

"New" Partitioned Data Set:
   Data set name . . . . . 'FMNUSER.TEMPLATE'
   Member  . . . . . . . . EXPORT1      Blank or pattern for member list
   Volume serial . . . . . _______      If not cataloged


Processing Options:

   Enter "/" to select option
   _  Batch execution                    _  Advanced member selection
                                         _  Skip member name list
Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 192. Load Module Compare: Entry panel

"Old" Partitioned Data Set:
You can use this group of fields to specify the "Old" load module to be compared. It contains:

Data set name
Specify the data set being a load module library (PDS). The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be blank.

Member
If you specified the name of a data set (PDS) without including a member name or name pattern in parenthesis in the **Data Set name** field, then you can use this field to specify the member name or a member name pattern.

Volume
Specify a volume serial number if the data set is not cataloged.

"New" Partitioned Data Set:
You can use this group of fields to specify the "New" load module to be compared. It contains:

Data set name
Specify the data set being a load module library (PDS). The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be blank.

Member
If you specified the name of a data set (PDS) without including a member name or name pattern in parenthesis in the **Data Set name** field, then you can use this field to specify the member name or a member name pattern.

Volume
Specify a volume serial number if the data set is not cataloged.

Note: Specification of the "New" load module depends on the specification of the "Old" load module. If the "Old" load module specifies one member, the "New" load module must point to one member. If the "Old" load module specifies a member name pattern, the "New" load module must use the same pattern or "*".

Processing Options
The processing options enable you to specify additional processing parameters. The options are:

Batch execution
Enter "/" to edit the JCL to run the function in batch.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

Skip member name list
Enter "/" to run without showing the member selection list.

Parent panels

- "Load module utility functions panel" on page 594

Child panels

- "Compare Utility: Load module options panel" on page 475

Equivalent functions

- "CLM (Compare Load Module)" on page 856
- "DSM (Data Set Compare)" on page 934

Related tasks and examples

- "Comparing data sets" on page 281

Load Module Information panel

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager             Load Module Information

 Load Library   FMN.LOAD
 Load Module    EXITINVO
      Linked on 2007.186 at 17:16:02 by PROGRAM BINDER 5695-PMB V1R7
            EPA 000000 Size 0000138 TTR 011C24 SSI          AC 00 AM 31  RM ANY

CSECT name        Type Address Size    RMODE AMODE Compiler 1 Date 1   Compil →
*                 *    _______ _______ *     *     *          *       *
<---+----10---+-> <--> <---+-> <---+-> <---> <---> <---+----> <---+--> <---+>
EXITINVO          SD   0000000 0000138  31   ANY   HLASM V1R5 2007.186
EXITINVO.EXITINVO  LD  0000000
****  End of data  ****




Command ===> ____________________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 193. View Load Module: Information panel

Load Library
The name of the load library for the load module.

Load Module
The name of the load module.

Linked on *date* **at** *time* **by** *program number*
The date and time the load module was link edited (bound) and the program number of the linkage editor or binder that was used.

EPA Entry point address of the load module.

Size The length (in hexadecimal) of the load module.

TTR The address of the member in Track and Record (TTR) format.

SSI The system status index (SSI) of the load module.

AC The authorization code (AC) of the load module.

AM The addressing mode (AMODE) of the load module.

RM The residency mode (RMODE) of the load module.

CSECT name
The section symbol or zap identification. This column can contain the following special values:

- **-PRIVATE** - indicates that the CSECT is unnamed (private).
- **-PSEUDOR** - indicates that the module contains definition statements for pseudo registers.
- **(BLANK)** - indicates that the common section (CM) is unnamed.

Type The section symbol type:
CM A common section definition
SD A section definition

PC A private section definition
LD A label definition
ZAP ZAPped or SZAPped. ZAP identifier and date ZAP applied

Address
The offset (hexadecimal) of the symbol in the load module.

Size The length (hexadecimal) of the section.

A/RMODE
AMODE/RMODE of the CSECT.

Compiler 1
The short name or number of the first compiler (if available).

Date 1 The date of the first compilation / ZAP applied (if available).

Compiler 2
A short name or the program number of the second compiler (if available).

Date 2 The date of the second compilation.

User data
User-provided data (identification, comments, and so on).

Parent panels

Child panels

Related tasks and examples

- "Manipulating your view of selection lists" on page 28

Load Module Information - Selection panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager              Load Module Information

Input:
   Data set name . . . . . 'FMN.LOAD'
   Member  . . . . . . . . FLMS7C     Blank or pattern for member list
   Volume serial . . . . . ______     If not cataloged


Processing Options:

   Order CSECTs by                     Output to
   1   1. Address                      1   1. Display
       2. Name                             2. Printer

   Enter "/" to select option
   _   YY/MM/DD date format (default: YYYY.DDD)
   _   Batch execution                 _   Advanced member selection
                                       _   Skip member name list

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 194. View Load Module: selection panel

Data set name
Specify the data set being a load module library. The data set name may

include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be blank.

Member
If you specified the name of a data set without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

For a member selection list, enter a blank or a pattern.

To select all members, enter an asterisk (*).

Note: For batch execution, blank selects all members and a pattern selects all matching members.

Volume serial
Serial number of the volume containing the data set. Required for data sets which are not cataloged.

Order CSECTs by
Use this option to specify the order in which you want the symbols listed.

Address (default)
Sorts the display by the offset (hexadecimal) of the symbol in the load module.

Name Sorts the display by symbol name.

Output to
Use this option to specify where you want the output to go.

Display
Displays the output to the terminal.

Printer
Prints the output to the printer. The printed output is created according to the print options you have specified.

YY/MM/DD date format
Enter "/" to request dates in YY/MM/DD format. YYYY.DDD is the default.

Batch Execution
Enter "/" to edit the JCL to run the function in batch.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

Skip member name list
Enter "/" to run without showing the member selection list.

Parent panels

Child panels

Related tasks and examples

Load module utility functions panel

The Load module utility functions panel allows you to display either the Load Module Information panel, or the Load Module Compare panel.

Panel and field definitions

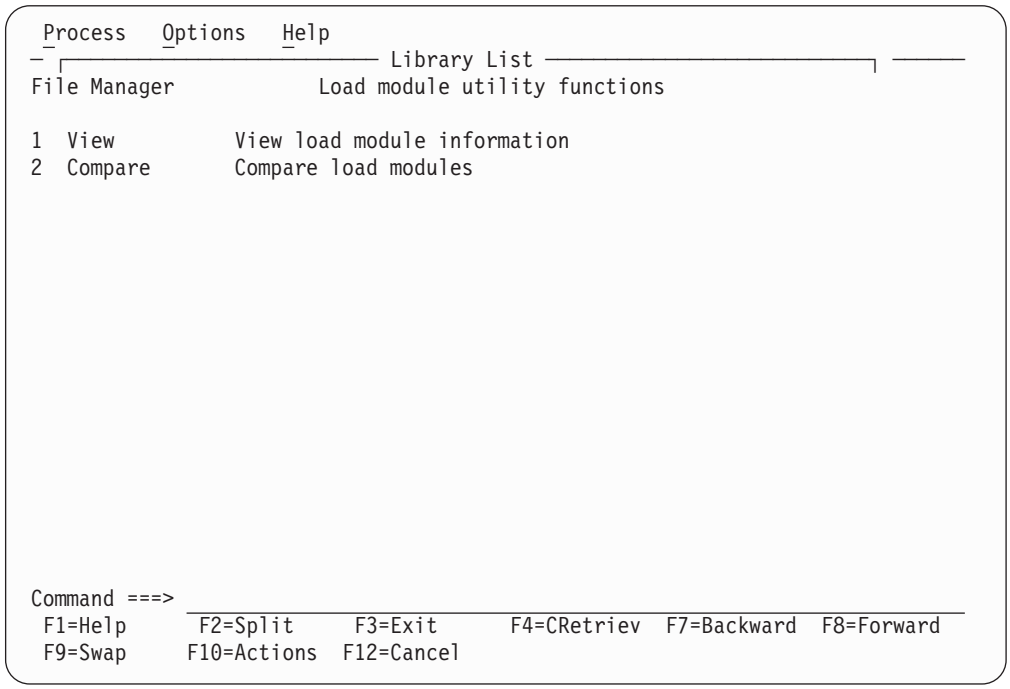


Figure 195. Load module utility functions panel

1 View
Displays the Load Module Information panel.

2 Compare
Displays the Load Module Compare panel.

Parent panels

- "Utility Functions menu panel" on page 703

Child panels

- "Load Module Information panel" on page 592
- "Load Module Compare - entry panel" on page 590

Related tasks and examples

- "Comparing load modules" on page 297

Map To panel

The Map To panel is used to select or create a copybook or template that can be used as the "To" template in a copy action or the "New" template in a compare action. The templates stores your manually-edited field mappings. The panel can be accessed by entering either the MC or MT commands on the Template Workbench.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
Map from FMN.SFMNSAM1(FMNCCPY)

CC Create template from copybook       EM Edit mapping
CM Create template from model          U  Update template from copybook
GM Generate corresponding map          GE Generate and edit mapping

To Copybook:
   Data set name . 'FMNUSER.COPYBOOK'_____________________________________
   Member  . . . . FMNCCPY___
To Template:
   Data set name . 'FMNUSER.TEMPLATE'_____________________________________
   Member  . . . . FMNCCPY_
Model Template:
   Data set name . ________________________________________________________
   Member  . . . . ________




Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Describe  F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

Figure 196. The Map To panel

Map from
The name of the copybook or template that contains the record descriptions for the "From" data set.

Map from commands
The Map from commands that you can use are listed at the top of the panel. You can position the cursor on the required command and press Enter or you can type them on the Command Line and press Enter.

Note: The RT and RC commands are only displayed and available when this panel is displayed by invoking one of the supported File Manager copy functions in advance-function mode. The command is not available when the Template Workbench is displayed using the AF function or a File Manager function other than a copy function.

These commands are:

CC The CC command is used to create a template from a copybook. You can edit the template before running a function with it. You can also save the template for later use.

You must specify the PDS member name of the copybook you want to use. The copybook you specify is validated by compiling it. File Manager includes the copybook in a shell program, and invokes the compiler to syntax-check the copybook, and to produce an ADATA file. If the compilation completes without errors, the information in the ADATA file is processed to create a template. If the compilation completes with errors, a pop-up menu is displayed. From the pop-up menu you can choose to:

- View the compilation listing.
- Abort the template creation process.

- Retry the compilation.

  Before you select this option, you should first view the compilation listing and correct any errors in the copybook. While you are viewing the compilation listing, you can use File Manager's split screen facility to swap to another session and use your editor to correct the errors in the copybook.
- Ignore the errors and proceed with creating the template.

  This option is only available if File Manager is able to create a template. Some compilation errors, such as warnings, might have no effect on the creation of the template. If you are unsure, you should view the compilation listing. If the compilation error is too severe, File Manager cannot create a template.

If File Manager is able to create a template and you provide the name of a sequential data set or PDS member in which the template is to be saved, File Manager saves it for you to use later with other functions. If you do not provide a template name, the template is kept in storage until the end of the current function. You can still use other Template Workbench commands to work with the template, but it is not automatically saved. However, if, at any time before the end of the current function, you decide to save the template you can specify a template name and save it using the SAVE command.

CM The CM command is used to create a new template by copying an existing template.

You must provide the data set name or PDS member name of the existing template. If you provide the name of a data set or PDS member in which the new template is to be saved, File Manager saves it for you to use later with other functions. If you do not provide a template name, the template is kept in storage until the end of the current function. You can still use other Template Workbench commands to work with the template, but it is not saved for later use. If, at any time before the end of the current function, you decide to save the template you can specify a template name and save it using the SAVE command.

EM The EM command is used to edit field mapping specifications in an existing template.

Field mapping allows you to change the layout of records when copying from one data set to another. The template you edit can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function. You can create a temporary template by using the CC command. (You can also create a temporary template by using either the GE, or the GM command, but you should only use these commands if you want File Manager to perform automatic mapping of fields with corresponding names.) Alternatively, you can let File Manager automatically create one for you. To do this, specify the member name of a COBOL copybook, but do not specify a template data set name or member name. Then, when you issue the EM command, File Manager automatically creates a temporary template.

When you edit field mapping in a template, a series of panels is displayed on which you can specify the following information:

- For a file containing multiple record types, which record types you want to map.
- Which fields in the "From" data set record descriptions map to which fields in the "To" data set record descriptions.

GE The GE command is used to automatically generate field mapping specifications using the GM command, and then invoke the EM command so you can review and, if necessary, modify the mapping. See the descriptions of these commands for details of how the mapping is done.

GM The GM command is used to automatically generate field mapping specifications in an existing template.

Field mapping allows you to change the layout of records when copying from one data set to another and to specify which fields are included when comparing data sets. The template can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function.

You can create a temporary template by using the CC command. Alternatively, you can let File Manager automatically create one for you. To do this, specify the member name of a COBOL copybook, but do not specify a template data set name or member name. Then, when you issue the GM command, File Manager automatically creates a temporary template.

When you issue the GM command, File Manager automatically maps record types and fields in the "From" template to record types and fields in the "To" template, as follows:

- Record types (COBOL level-01 data items) are firstly paired by name. Each record type in the "From" template is paired with a record type with the same name in the "To" template, if one exists.
- Once all record types with matching names have been paired, any unpaired record types are paired by position in the template. For example, if the first record type in the "From" template is unpaired, and the first record type in the "To" template is unpaired, the two record types are paired. Checking then continues with the second record type in each template, and so on. Pairing stops when there are no more record types in either the "From" or "To" template.
- Once record types have been paired, fields in the paired record types are mapped using much the same rules as the COBOL MOVE CORRESPONDING statement. A field in a "From" record type is mapped to a field in a "To" record type if both fields have the same name and, in the case of a field that is an elementary field in a group field, the respective group fields have the same names.

RC The RC command is used to run the current function using a temporary template created from a copybook.

The temporary template cannot be edited before the function is run, and is not saved. Use the CC command if you want to edit or save the template.

You must specify the PDS member name of the copybook you want to use, and the data set name or PDS member name of the

template you want to update. The copybook you specify is validated by compiling it. For a description of this process, and information about what you can do if errors occur, see the description of the CC command.

This command is only available when the Template Workbench is displayed by invoking one of the supported File Manager copy functions in advance-function mode. The command is not available when the Template Workbench is displayed using the AF function or a File Manager function other than a copy function.

RT The RT command is used to run the current function with an existing template.

The template can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function using the CC command.

This command is only available when the Template Workbench is displayed by invoking one of the supported File Manager copy functions in advance-function mode. The command is not available when the Template Workbench is displayed using the AF function or a File Manager function other than a copy function.

U The U command is used to update an existing template from a copybook.

You should use this function when you have made minor changes to the copybook, and you want these changes reflected in the template you previously created from the copybook. If you have made significant changes to the copybook, the update process might not produce the results you expect, so you should instead use the CC command to recreate the template.

Minor changes that the update process supports include:

- Changing field names without changing field data types
- Changing field data types without changing field names
- Changing the order of fields in a record
- Deleting unreferenced fields
- Inserting new fields
- Changing record length
- Changing the number of occurrences of fields in an array (table)

Major changes that might cause the update process to produce unwanted results include:

- Changing field names and field data types
- Changing field names and the order of fields in a record

You must specify the PDS member name of the copybook you want to use, and the data set name or PDS member name of the template you want to update. The copybook you specify is validated by compiling it. For a description of this process, and information about what you can do if errors occur, see the description of the CC command.

To Copybook

The name (data set and member) of the copybook that contains the record descriptions for the "To" data set.

To Template
The name (data set and, if required, member) of an existing template you want to run with or edit, or a new template you want to create. This template contains the record descriptions and field mappings for the "To" data set.

Model Template
The name (data set and, if required, member) of an existing template you want to use as a model for creating a new template.

Parent panels

Child panels

Related tasks and examples

Member Selection panel

The Member Selection panel provides a list of PDS(E) members, from which you can select the members to be included in your current process.

This panel is displayed:

- When you have specified a PSD(E) data set on an Entry panel and either left the Member field blank or entered a member name pattern.
- For CA-Panvalet directory lists
- For other library management system library directory lists

Panel and field definitions

```
 Process   Options   Help
 File Manager              Edit Member Selection            Row 00001 of 00416
DSNAME CARLAND.FMN.ADCU
       Name     Prompt   Alias-of Lib   Size    Created    Changed           →
       *        *        *        *     ______  *          *
______ zzz
______ AABBCC
______ ABC
______ ABCDEF                                41 2010/08/02 2010/08/02 11:01:01
______ ABEND0C7
______ AB12345
______ ACTFX
______ ACTFXS
______ ACTFXSC
______ ACTFXU
______ ACTFXUC
______ ALIAS
______ AMASPZAP                              10 2010/08/02 2010/08/02 10:59:08
______ AMBLIST                               10 2003/04/01 2009/12/23 16:55:59
______ APARDEF
______ ARRAYNUM
______ ARRAY1
______ ARRAY2
______ ARRRAY1
______ ASMADOP                               17 2003/01/22 2003/01/22 14:23:01
______ ASMCOMPO                              31 2001/07/06 2002/10/21 16:03:26
______ ASMDASM                               13 2002/04/04 2004/04/13 14:56:23
______ ASMDECK                               17 1999/06/24 2006/01/12 14:39:18
Command ===>                                                        Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 197. Member Selection panel

The fields displayed vary depending on the directory list being displayed. There are four types of displays:

- PDSs with formatted records
- PDSs with unformatted records (such as loadlibs)
- CA-Panvalet directory lists
- Other library management system library directory lists

The common fields displayed (for all displays) are described below, followed by the specific fields for each type. Other library management system library directory lists are determined by the customer-provided interface and are not listed here.

(Prefix area)
Area in which you can enter one of the following prefix commands:

Browse
Browse the member.

Copy Copy the member and any related alias members.

Delete Delete the member and any related alias members.

Edit For member lists restricted to showing only templates, IMS views, or extract criteria, the corresponding template, view, or criteria edit is invoked. For all other member lists, it edits the member. View is substituted for members managed by other library systems.

G Synonym for RESET prefix command.

Jcl This command is only active for loadlib member lists and generates the link edit JCL for the load module.

Move Move the member and any related alias members.

Print Invoke the File Manager Print utility to print the member.

REName
Rename the member.

RESet Invoke the File Manager Reset ISPF Statistics function to reset or delete the ISPF statistics for the member.

Select /
Select a member. Depending on the function from which this panel has been invoked, results in one of these actions:

- A single member being selected and returned to the calling routine. (Note: subsequent selections are ignored)
- Toggling of the selected value in the prompt field.
- Selection of a member to be processed by the function that invoked the member list.

Note: This command is not valid when the panel is displayed as the result of the MEMBER prefix command.

SUbmit
Submits the member to the JES internal reader.

Update
This command is only active for member lists restricted to showing base templates only and invokes the source definition edit/update process for a selected template.

View View the member if the member is not a load module. If the member is a load module, the load module CSECT information is displayed.

Name The name of the member.

Prompt
A dynamic area that displays the last action that occurred for the listed member.

Alias-of
Displayed for PDS directories and shows the alias names. If a broken alias is encountered, “no-alias” is displayed.

Lib The relative number of the library (as listed on the Library List panel). This column only appears if more than one library directory has been processed.

ID The User ID of the person who last changed the member.

Columns for formatted PDS members:

Size For members stored with ISPF statistics, the number of statements in the member.

Created
The date created, in YY/MM/DD format.

Changed
The date and time, in YY/MM/DD-HH:mm:SS format, on which the member was last changed.

INIT The initial number of records in the member.

MOD The current number of records in the member

VV.MM
The version number of the member.

Columns for unformatted PDS members:

Size The hexadecimal value for the load module size.

AC Authorization code.

AM AMODE binder value.

RM RMODE binder value.

Attributes
Attribute values determined by binder.

TTR TTR of first block of text.

EPA Entry point address associated with member name or with alias name if alias.

SSI SSI information word.

Columns for CA-Panvalet members:

Lvl The member level.

Stat The member status (such as T-TEST, P-PROD and so on).

User User code.

Accessed
Last access date.

Changed
Last changed date.

Size Number of statements.

Blks Number of blocks.

Note: Some utilities (within ISPF, z/OS or third party products) do not update the statistics kept for the Size, Created, Changed and ID fields. If one of these utilities has recently changed the member, those fields are blank.

Available commands

The following primary commands are available:

- "BOTTOM primary command" on page 742
- "DOWN primary command" on page 760
- "FIND/FX primary command" on page 769
- "HEX primary command" on page 780
- "LEFT primary command" on page 784
- "LOCATE primary command" on page 786
- "REFRESH primary command" on page 804

- "RESET primary command" on page 806
- "RFIND primary command" on page 809
- "RIGHT primary command" on page 810
- "SORT primary command" on page 818
- "TAILOR primary command" on page 823
- "TOP primary command" on page 824
- "UP primary command" on page 826

Parent panels

- "Browse Entry panel" on page 459
- "Edit Entry panel" on page 541
- "Data Create Utility panel" on page 515
- "Copy From panel" on page 491
- "Copy To panel" on page 497
- "Print Utility panel" on page 620
- "Find/Change Utility panel" on page 574
- "AFP Print Browse panel" on page 442
- "Load Module Information panel" on page 592
- "Compare Utility: "Old" and "New" panels" on page 471
- "Template Workbench panel" on page 698
- "Template Build Utility panel" on page 688

Child panels

- The display panel that would normally result from your starting Entry panel.

Related tasks and examples

- "Specifying a data set and a member name" on page 16
- "Selecting a range of PDS(E) members" on page 34
- "Manipulating your view of selection lists" on page 28

Memory Browse panel

The Memory Browse panel displays your user storage in dump format.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                    Memory Browse
Browse address 00000000_                                 Search limit *_______

Location   Offset  -----------------Hex------------------   -------Char-------
00000000        0  040C0000  81342410  00000000  00000000   *....a..........*
00000010       10  00FCD208  00000000  070C2000  8138692A   *..K........a...*
00000020       20  078D2000  80CDF22E  040CE000  80FEEE6E   *......2...\....>*
00000030       30  078D1000  9854146C  070E0000  00000000   *....q..%........*
00000040       40  00000000  00000000  00000000  00FCD208   *..............K.*
00000050       50  00000000  00000000  040C0000  8132FEA8   *............a..y*
00000060       60  040C0000  80FFA080  00080000  9F56E888   *.............Yh*
00000070       70  00080000  9F56F948  040C0000  81331700   *......9.....a...*
00000080       80  00000000  00001005  00020078  00040016   *................*
00000090       90  7F7DD001  00000000  00000000  00000000   *"'}............*
000000A0       A0  00000000  0133E308  00000000  00000000   *......T........*
000000B0       B0  00000000  00000000  000102FD  0232B3A8   *..............y*
000000C0       C0  28000000  00000000  00000000  00000000   *................*


Command ===> ____________________________________________________ Scroll PAGE_
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Previous F11=Next     F12=Cancel
```

Figure 198. Memory Browse panel

Browse address
Starting address of the data you want to browse.

Search limit
Limits the search to a specified number of bytes. You enter a hexadecimal number or * (the default). By default the entire storage is scanned, in pieces of 64KB each.

Location
Shows the hexadecimal location of the bytes displayed.

Offset Shows the offset of the 16 bytes displayed relative to the selected address for Location.

Hex Displays the hexadecimal representation of the storage data.

Char Displays the storage data in character format.

The following primary commands are available on this panel:
- BACK - this command is unique to the Memory Browse panel.
- CLIPBOARD - this command is unique to the Memory Browse panel.
- FIND - see “FIND/FX primary command” on page 769
- NEXTPAGE - this command is unique to the Memory Browse panel.

The BACK primary command restores the previous browse address. It returns you to offset zero of the current browse address or, if the offset is zero, it restores the browse address saved in the previous position of the pointer chain.

The CLIPBOARD primary command displays the data currently saved on the clipboard using the PUT command in an Edit session (see “Copying data to and from a clipboard” on page 121).

The length of the clipboard data is stored beginning with offset X'C'; the data itself is stored beginning with offset X'10'.

Note: The CLIPBOARD primary command only displays data saved on the clipboard if Memory Browse is invoked from an Edit session (during which you have saved data to the clipboard using the PUT command).

The NEXT primary command sets a new browse address and saves the current browse address. Either place the cursor onto a pointer value visible on the screen, or enter the new address into the **Browse address** field, then enter NEXT (or use a corresponding function key). To go forward to a previously saved address, check that your cursor is not on a data field, then enter NEXT (or use a corresponding function key). File Manager saves up to 64 browse addresses in a chain.

Use the scroll commands to move the display window over the user storage. For details, see "Scrolling to see data" on page 77.

Parent panels

Child panels

Related tasks and examples

Non-VSAM Allocate panel

This panel is used to define a catalog entry for a non-VSAM file type. It allocates the data set as it creates the catalog definition.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                 Non-VSAM Allocate

Non-VSAM Data Set:
   Data set name . . 'USERID.TESTCATS.DATA2'
                                                                    More:     +
Specify volume serial, SMS class names, or leave blank for defaults:
   Device type . . .  _____         generic unit or device address
   Volume serial . .  ______
   Data class  . . .  ________      leave blank for default
   Storage class . .  ________      leave blank for default
   Management class   ________      leave blank for default

Space Requirements:
   Space unit  . . .  CYL           BLKS, TRKS, CYLS, KB, or MB
   Primary units . .  1             quantity of above units
   Secondary units .  5             quantity of above units
   Directory blocks   0             leave blank for SMS default
   Record format . .  VB            if new format: U,F,V, or D, with B,S,A,M
   Record length . .  8000
   Block size  . . .  27998         physical output block size
   Library type  . .  LIBRARY       LIBRARY, PDS, or blank for default
   Expiration date .  ________      yyyy.ddd, blank for default
Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 199.

The fields in this panel are identical to those in the "Allocate (Two) panel" on page 451.

Parent panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Child panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Related tasks and examples

- "Defining a new Catalog Entry" on page 319

Non-VSAM Association Information panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager          Non-VSAM Association Information

NONVSAM Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.NVSAM1'
   Catalog ID . . . . 'CATALOG.UCATAPC'
                                                                        More:
NONVSAM Associations:
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS1'
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS2'
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS3'
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS4'
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS5'
   Alias  . . . . . . 'SYATES.UNITTEST.RFM0189.NVSAM1.ALIAS6'
                      ****  End of data  ****




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Info      F7=Up
 F8=Down      F9=Swap     F10=Actions   F12=Cancel
```

Figure 200. Non-VSAM Association Information panel

Parent panels

Child panels

Related tasks and examples

Non-VSAM Define panel

This panel is used to define a catalog entry for a non-VSAM file type. It does not allocate the data set.

Panel and field definitions

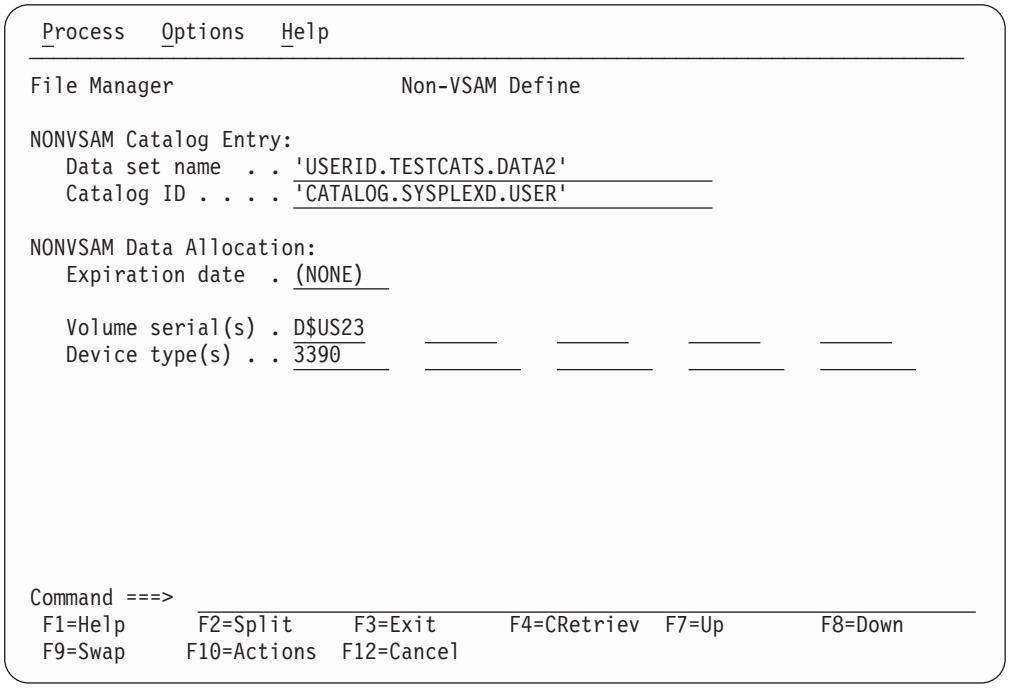


Figure 201. Non-VSAM Define panel

Data set name
The name of the catalog entry, from one to 44 characters. For DEFINE and ALTER commands, enter the new name.

Catalog ID
Name of a catalog that contains the entry. Default: System catalog search order.

Expiration date
Expiration date in the form YY.DDD or YYYY.DDD, where (YY)YY denotes the year and DDD denotes the number of the day from 1 to 366.

Leave it blank if an expiration date is not desired.

Volume serial(s)
Serial number of a DASD or tape volume to contain the non-VSAM data set.

Device type(s)
Specifies the device type of one or more volumes containing the data set. Enter an IBM device type or a generic device name that is supported by your system.

Parent panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Child panels

- "Catalog Services panel" on page 464
- "Catalog Services Data Set List panel" on page 466

Related tasks and examples

- "Defining a new Catalog Entry" on page 319

Non-VSAM Entry Detail panel

When you invoke the alter, define, or information service or line command for a non-VSAM catalog entry, detailed catalog information is displayed for the selected entry. For most fields you can view a description of the field by placing the cursor on the field and pressing F1 (Help).

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                Non-VSAM Entry Detail

Catalog Entry:
   Data set name  . . 'SYATES.COMPARE'
   Catalog ID . . . . 'CATALOG.USER3.SYSPLEXD'

Additional information available with Extents(F11) and ASsocs(F6) commands.

General Data:
   SMS managed . . . . Y              Current Allocations:
   Management class  . PRIMARY           Allocated tracks . . 5
   Storage class . . . PRIMARY           Allocated extents  . 1
    Volume serial  . . D$US24            Maximum dir blocks . 40
    Device type  . . . 3390
   Data class  . . . . **None**       Current Utilization:
    Organization . . . PO                KB used  . . . . . . 287
    Record format  . . FB                Tracks used  . . . . 5
    Record length  . . 80                % Utilized . . . . . 100
    Block size . . . . 27920             Number of members  . 64
    1st extent trks  . 5
    Secondary trks . . 19
    Data set name type PDS
   Last backup date  . 0000.000.0000
   Creation date . . . 2002/02/27 (058)
   Expiration date . . ***None***
   Referenced date . . 2005/09/09 (252)
   Change indicator  . YES

Extended Attributes:
   Data attributes    Extended  . N      Compressed  . . . N
                      Striped . . N      Stripe count  . . -

Associations:
   GDG base . . . . .
   Alias  . . . . . .
Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=Volumes   F6=Assocs
 F7=Up        F8=Down      F9=Swap     F10=Actions  F11=Extents  F12=Cancel
```

Figure 202.

Parent panels

Child panels

Related tasks and examples

- "Non-VSAM Extent Information panel" on page 610

Non-VSAM Extent Information panel

The Non-VSAM Extent Information panel displays the extents of a disk data set.

To display the Non-VSAM Extent Information panel, enter the EX line command on the Data Set List panel.

To locate specific data, use the LOCATE primary command.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager             Non-VSAM Extent Information             Line 1 of 430

 Data set name: GREGCZ.X1KVSM06.FILLE1
 Total volumes: 10  Extents: 430   Tracks: 430

                  Begin - ABSOLUTE - End            Begin - RELATIVE - End
  Ext   Trks   Cyl-hd Rn...n   Cyl-hd Rn...n   Cyl-hd Rn.....n   Cyl-hd Rn.....n

 VOLSER: D$ST04(1)  Extents: 43  Tracks: 43 -------------------------------
    1      1   550  4   8254   550  4   8254     0  0        1     0  0        1
    2      1   550  6   8256   550  6   8256     0  1        2     0  1        2
    3      1   550  7   8257   550  7   8257     0  2        3     0  2        3
 ...
   43      1   553  2   8297   553  2   8297     2 12       43     2 12       43

 VOLSER: D$ST01(2)  Extents: 43  Tracks: 43 -------------------------------
   44      1   550  6   8256   550  6   8256     2 13       44     2 13       44
   45      1   550  8   8258   550  8   8258     2 14       45     2 14       45
   46      1   550  9   8259   550  9   8259     3  0       46     3  0       46
 ...
  387      1   552  2   8282   552  2   8282    25 11      387    25 11      387

 VOLSER: D$ST05(10)  Extents: 43  Tracks: 43 ------------------------------
  388      1   900  5  13505   900  5  13505    25 12      388    25 12      388
  389      1   900  7  13507   900  7  13507    25 13      389    25 13      389
  390      1   900  8  13508   900  8  13508    25 14      390    25 14      390
 ...
  428      1   903  1  13546   903  1  13546    28  7      428    28  7      428
  429      1   903  2  13547   903  2  13547    28  8      429    28  8      429
  430      1   903  3  13548   903  3  13548    28  9      430    28  9      430
******************************** End of data ********************************
 Command ===> _____________________________________________________ Scroll CSR
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 203.

Parent panels

- "Catalog Services Data Set List panel" on page 466

Child panels

Related tasks and examples

OAM Functions panel

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                    OAM Functions

 1  Directory     Display or print object directory
 2  Browse        Browse object
 3  Print         Print object
 4  Update        Update object
 5  Erase         Erase object
 6  Copy          Copy object









 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 204. OAM Functions panel

Parent panels

"Primary Option Menu panel" on page 615

Child panels

- "Object Directory List panel"
- "Object Browse panel"
- "Object Print panel"
- "Object Update panel"
- "Object Erase panel"
- "Object Copy Functions panel"

Related tasks and examples

- Chapter 10, "OAM Functions," on page 383

Path Entry Detail panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Path Entry Detail

PATH Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.ESDS3.PATH1'
   Catalog ID . . . . 'CATALOG.UCATAPC'

PATH Associations:
   Creation date  . . 2003.351      Expiration date. . (NONE)

   AIX  . . . . . .
     Data component .
     Index component

   Cluster  . . . . . 'SYATES.UNITTEST.RFM0189.ESDS3'
     Data component . 'SYATES.UNITTEST.RFM0189.ESDS3.DATA'
     Index component



Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 205. Path Entry Detail panel

Parent panels

Child panels

Related tasks and examples

Personal Data Set List panel

The Personal Data Set List panel displays the contents of a data set list. You can edit the list by typing in new details, by overtyping the existing information, or by entering the E line command to display the Edit Personal Data Set List panel which provides more edit options.

You can display an alternative form of this panel with the LISTVIEW primary command that shows a third line for each entry where you can view or enter a description for the entry.

Panel and field definitions

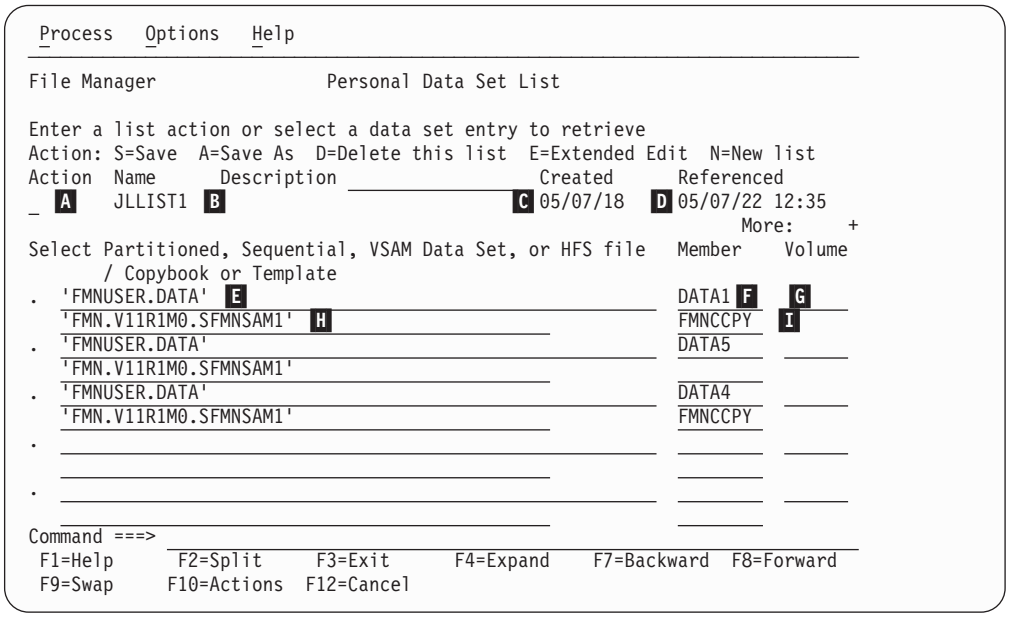


Figure 206. Personal Data Set List panel

A Line command entry field. Valid values are:
- **S** Save. Saves any changes made to the personal data set list.
- **A** Save As. Saves current list entries to a new list.
- **D** Delete. Deletes the list.
- **E** Extended Edit. Invokes the personal data set list edit dialog.
- **N** New. Create a new list.

B Name of this data set list

C Date this data set list originally created

D Date and time this data set list referenced

E Data set name

F Member name

G Volume serial number

H Data set name of copybook or template

I Member name of copybook or template

Available commands

The following primary commands are available on this panel:
- LISTVIEW

Parent panels

- "Personal Data Set Lists panel" on page 614

Child panels

- "Edit Personal Data Set List panel" on page 543

Equivalent functions

- None.

Related tasks and examples

- "Working with data set lists" on page 235

Personal Data Set Lists panel

The Personal Data Set Lists panel displays a list of all your data set lists, including the reference list, REFLIST, which is maintained by File Manager.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager              Personal Data Set Lists            Row 1 to 3 of 3
Active: JLLIST1
Action: O=Open  A=Save As  D=Delete  E=Edit  N=New list

   Name             Description                    Created   Referenced
_  REFLIST          Last 30 referenced data sets   --------  -------- -----
_  JLLIST1                                         05/07/18  05/07/22 09:24
_  JLLIST2                                         05/07/21  05/07/21 16:43
****************************** Bottom of data ********************************









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 207. Personal Data Set Lists panel

Command entry field
Line command entry field. Valid values are:

- **O** Open. Opens the data set list.
- **A** Save As. Displays a save confirmation panel where you can enter the name you want to call the data set list and, optionally, a list description.
- **D** Delete. Deletes the data set list.
- **E** Edit. Displays the Edit Personal Data Set List panel.
- **N** New. Displays a new Personal Data Set List panel with empty entries.

Name Name of the data set list

Description
Description of the data set list.

Created
Date this data set list originally created

Referenced
Date and time this data set list referenced

Child panels

- "Personal Data Set Lists panel"

Equivalent functions

- None.

Related tasks and examples

- "Working with data set lists" on page 235

Primary Option Menu panel

The Primary Option Menu panel is the starting point for all File Manager operations.

Panel and field definitions

```
1  Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Primary Option Menu
 2                                                                  3
0  Settings      Set processing options                User ID . : USERID
1  View          View data                             System ID : MVS8
2  Edit          Edit data                             Appl ID . : FMN
3  Utilities     Perform utility functions             Version . : 6.1.0
4  Tapes         Tape specific functions               Terminal. : 3278
5  Disk/VSAM     Disk track and VSAM CI functions      Screen. . : 1
6  OAM           Work with OAM objects                 Date. . . : 2006/09/24
7  Templates     Template and copybook utilities       Time. . . : 14:52
8  HFS           Access Hierarchical File System
9  Websphere MQ  List, view and edit MQ data
10 CICS          FM/CICS
X  Exit          Terminate File Manager









Command ===> ____________________________________________________________
4 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 208. File Manager Primary Option Menu panel

1 Action Bar

The Action Bar contains a number of pull-down menus. These menus are available throughout File Manager.

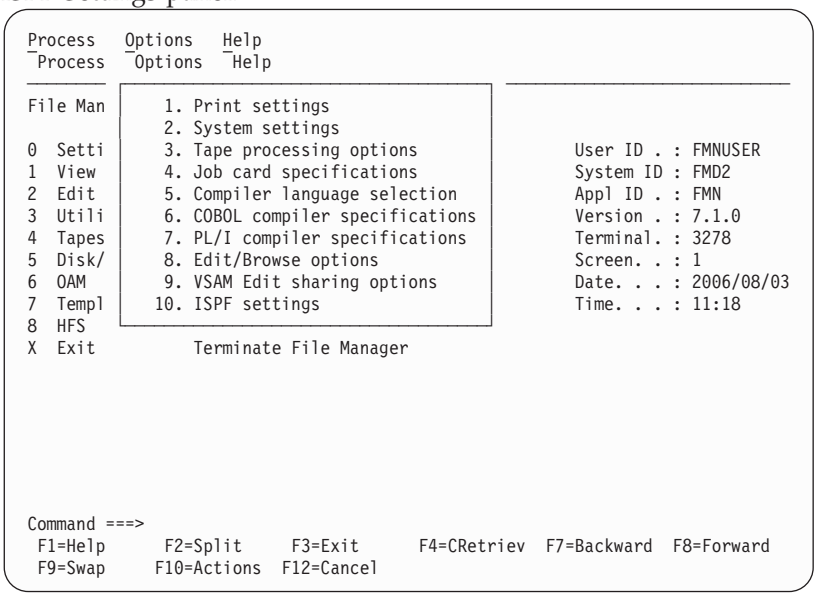
Process

The Process pull-down menu lists the processes available on your current panel. These options may change, depending upon which panel is being viewed.

```
 Process   Options   Help
┌──────────────────────┐ ──────────────────
│ _ 1. Exit File Manager │   Primary Option Menu
└──────────────────────┘
 0  Settings      Set processing options
⋮
```

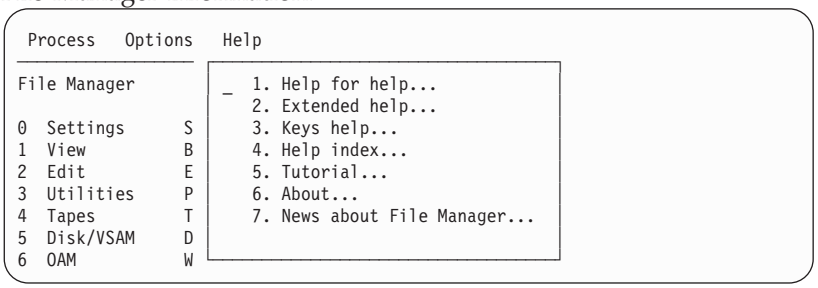
Options

The Options pull-down menu lists the choices available from the Set Processing Options panel, as well as providing access to the ISPF Settings panel.



See "Compare Utility: Options panel" on page 479 for details about options 1 to 8. See the *ISPF User's Guide Volume 1* for details about the ISPF Settings panel.

Help The Help pull-down menu lists the various ways in which you can enter the Tutorial Help system, as well as providing access to more File Manager information.



1. Help for help
Displays the Help panel for the Tutorial Help system.

2. Extended help
Displays the Tutorial Help panel associated with the current File Manager panel (equivalent to pressing F1 from the File Manager panel). When on the Primary Options menu, this is the Tutorial Help Table of Contents panel.

3. Keys help
Displays a panel that provides help for the Function Keys that are active on the current File Manager panel.

4. Help index
Displays the Help Index.

5. Tutorial
Displays the Tutorial Help Table of Contents panel.

6. About
Displays the File Manager version and release information in a pop-up window.

7. News about File Manager
Displays a panels providing general information about the current File Manager version/release.

2 Primary Options
Lists the Primary Options available within File Manager. Selecting an option displays an Entry panel or a Menu panel for the associated function.

3 Status Area
Displays your session information.

4 Function Keys
Lists the Function Keys that are active within the current panel.

Parent panels

ISPF Primary Options Menu - see "Chapter 1. Primary Options Menu" in the *ISPF User's Guide Volume I*.

Child panels

- "Set Processing Options panel" on page 661
- "Browse Entry panel" on page 459
- "Edit Entry panel" on page 541
- "Utility Functions menu panel" on page 703
- "Tape Specific Functions panel" on page 677
- "Disk/VSAM Data Functions panel" on page 523
- "OAM Functions panel" on page 611
- "Template and Copybook Utility functions panel" on page 687
- "Access Hierarchical File System panel" on page 438
- "WebSphere MQ Functions panel" on page 728
- "Primary Option Menu panels" in the *File Manager User's Guide and Reference for CICS*

Equivalent functions

- None.

Related tasks and examples

"Starting and exiting File Manager" on page 9

Print Audit Trail panel

Panel and field definitions

```
 Process   Options   Utilities   Help
 ------------------------------------------------------------------------------
File Manager                 Print Audit Trail

Audit Trail:
   Data set name . . . _____________________________________________
   Description . . . . ____________________________________

Processing Options:
   Enter "/" to select option           Formatted display options
   _  Formatted print                   2  1. Hex Format
   _  Print only changed fields            2. Non-display Hex
   _  Keep data set after printing      /  Highlight changes
   _  Browse report                     /  Show key fields
   _  Batch execution

Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 209. The Print Audit Trail panel

Formatted print
Displays report entries in formatted character representation, rather than hexadecimal format.

Print only changed fields
Limits the amount of printed lines.

Keep data set after printing
Whether to keep or delete the data set after it is printed.

Browse report
The audit trail report is displayed immediately.

Note: The contents of the audit trail report are transferred to a SYSOUT class as specified by the **PRINTOUT** print option on the Set Print Processing Options panel. (For details about setting this option, see "Setting your Print Processing Options" on page 299.) To browse the audit trail report, set the **PRINTOUT** print option to SYSOUT=*c*.

Batch execution
To generate JCL for printing the audit trail report using a batch job. For command syntax, see "AUD (Print Audit Trail Report)" on page 834.

Hex Format
To produce an UPDOWN hexadecimal display below the standard field display.

Non-display Hex
To produce an UPDOWN hexadecimal display below the standard field display only for fields that contain non-displayable \ characters.

Highlight changes
Highlight the changed fields. An asterisk is placed to left of the before data to indicate the field has been changed.

Show key fields
To display key fields even when **Print only changed fields** is selected. A "K" is printed to the left of key field names. For a KSDS data set, a key field is any elementary field that intersects or is contained in the key area.

Parent panels

Child panels

Related tasks and examples

Print Options panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                    Print Options

Press ENTER to proceed.

Formatted print options:

   /  Field reference number             /  Field type and length values
   /  Picture clause                     /  Start location
   /  Structure                          /  Left justify numeric fields
   /  Redefined fields                   /  Print record length

Other print options:

   /  Ignore record length mismatch

 Command ===> ____________________________________________________________
  F1=Help        F2=Split       F3=Exit        F4=Resize
  F5=Ex-help     F6=Keyshelp    F7=PrvPage     F8=NxtPage
```

Figure 210. Print options panel

Formatted print options:

Field reference number (REF)
Changes the session default to show the field reference (for example, #2) next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the REFS command.

Field type and length values (TYPE)
Changes the session default to show the field data type and length next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the TYPE command.

Picture clause (PIC)
Changes the session default to show the Picture clause when in SNGL display or print mode. Also shows the length and scale (if non zero) for binary and packed fields and the bit length for bit fields. In an editor session, this display can also be turned on or off temporarily, by using the PIC command.

Start location (SLOC)
Changes the session default to show the start location of each field, when in SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the SLOC command.

Structure (STR)
Changes the session default to show the full structure of the template when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the STR command.

Left justify numeric fields (JUST)
Changes the session default to show numeric fields as left justified, when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the JUST command.

Redefined fields (RDF)
Changes the session default to show the redefined information as part of the field name, when in SNGL display or print mode and display or print the redefined fields when in SNGL or TABL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the RDF command.

Print record length
Turns on or off the display of record lengths in TABL and SNGL formats. In an editor session, this display can also be turned on or off temporarily, by using the RECLEN command.

Ignore record length mismatch
Records failing the selection process because of length mismatch with template layouts) are considered selected for processing if they meet other selection criteria.

Parent panels

- "Print Utility panel"

Child panels

Related tasks and examples

Print Utility panel

The Print Utility lets you print sequential, partitioned data set members, VSAM files, or HFS files in a selected format.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Print Utility

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name  . . 'FMNUSER.EXPORT'                                  +
   Member  . . . . . . . . *          (Blank or pattern for member list)
   Volume serial . . . . . ______     (If not cataloged)
   Start key . . . . . . . ___________________________ key or slot
   Skip  count . . . . . . ________    Include _________ Repeat skip ________
   Print count . . . . . . ALL_____    number of records to be printed

Copybook or Template:
   Data set name . . . . . ____________________________________________
   Member  . . . . . . . . ________    (Blank or pattern for member list)

Processing Options:
   Print Option      Copybook/template/access      Enter "/" to select option
   1  1. Char        3  1. Above                   _  Edit template   _ Type (1,2,S)
      2. Hex            2. Previous                _  Batch execution
      3. Sngl           3. None                    7  Use proc ________
      4. Tabl           4. Physical Blk/CI Access  _  REXX member selection: P
   ISPF Packing         5. Create dynamic          _  Advanced member selection
   1  1. Unpack      MQ Processing                 _ Additional print options
      2. None        _ Include Descriptors         _ Binary mode, reclen _____
      3. Skip



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 211. Print Utility panel

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume
Specify a volume serial number if the data set is not cataloged.

Start key
The starting key or slot location for a VSAM file

Skip count
The number of records to be skipped before processing begins

Include
Represents the number of physical records that to be included in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Range: 0-9999999. Default: 0.

Repeat skip
Represents the number of physical records to be skipped in a repeating sample from a data set. Both **Include** and **Repeat Skip** fields must have non-zero values for sampling to take effect.

Range: 0-9999999. Default: 0.

Print count
The number of records to be printed

Copybook or Template: Data set name and Member
These fields identify either a copybook or a template to be used to select records and format the print output. If a copybook is specified, it is compiled to create a temporary template that can optionally be saved if edit was requested.

Print Option
The format of the printed output:

CHAR
Records are printed in character format.

HEX Records are printed in hexadecimal format. The DUMP processing option on the Set Processing Options panel determines how HEX data is printed. See "Set Processing Options panel" on page 661.

SNGL Single record format (one field on each line of the display). The left-hand side of the printout contains the field or header names. The right-hand side contains the field values formatted according to the field type. A template is required for this option.

In this print format, additional field information (redefined fields, field reference number, field type and length values, picture clause, start location, structure) can be printed depending on the editor options. Numeric fields can be left-justified depending on the options.

TABL Tabular format (fields printed across the display). The records are displayed in columns with field or heading names as column headers and the data formatted according to the field types. In this format, the display might be truncated depending on the current print width. A template is required for this option.

Copybook/template/access
Use this option to specify if you want to use a template, or how records are to be printed.

Above File Manager uses the template specified on the panel. (If you specified a copybook, then File Manager compiles the copybook into a template before use.)

Previous
File Manager uses the last template you used to process this data

set. If you have not previously used a template for this data set, a message is displayed and processing continues without using a template.

None File Manager does not use a template. SNGL and TABL print formats are not available.

Physical Blk/CI Access
Records are printed by physical block (non-VSAM) or by control interval (VSAM).

Create dynamic
File Manager creates a dynamic template. You can now define fields and field creation attributes.

MQ processing

Include Descriptors
If this is selected (with the / character), the message descriptor data returned by the GET MQ API is added as a prefix to the record data, allowing this data to be examined in conjunction with the message data.

Otherwise only the message data is available for processing.

Edit template
File Manager opens the template you have specified for editing, before processing the panel.

Note: For details about using templates, see Chapter 4, "Creating and editing templates," on page 127.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Batch execution
File Manager generates JCL to run the function in batch and then displays the JCL for editing. Use Set Batch Job Card Information (option 0.4) to tailor the default JOB card that File Manager uses to generate the JCL. See "Set batch job card information (option 0.4)" on page 45.

Use proc
Use this option to specify a procedure to search and change the data, rather than using the Find and Change commands. When you select the option, you can perform either of these actions

- Choose to create a temporary procedure for one-time use by entering a single asterisk (*). File Manager displays an Edit panel, in which you can create the procedure.
- Specify the name of the member containing the procedure you want to use, The member must belong to the PDS allocated to ddname FMNEXEC. You can enter any of the following:
  - The name of the member.

– A member name pattern (other than a single *) to list all matching members. You can then select the required member by entering an S in the **Sel** field. A member name pattern can consist of any characters that are valid in a member name and the following two special pattern characters:

  asterisk (*)
  Represents any number of characters. As many asterisks as required can appear anywhere in a member name. For example, if you enter *d*, a list of all members in the data set whose name contains "d" is displayed.

  percent sign (%)
  A place-holding character representing a single character. As many percent symbols as necessary can appear anywhere in a member name. For example, if you enter %%%%, a list of all members in the data set whose name is four characters in length is displayed.

Note: If you select this option but leave the **Use proc** member entry field blank, File Manager displays a member name list. You can then select the required member by entering S in the **Sel** field.

(Also, see "Supplying a procedure when using a File Manager panel" on page 417.)

REXX member selection
Enter "/" to use a REXX procedure to determine if a member should be copied.

If this option is selected, the **Use Proc** option must also be selected and a REXX procedure supplied. You can optionally specify the default action to be applied in the event that the nominated procedure fails to determine if a member in the input data set should be copied. The default action is determined by entering a P (PROCESS MEMBER) or D (DROP MEMBER) in the adjacent field. If no default is provided, P is assumed.

Advanced member selection
Enter "/" to specify a range of members to be selected rather than a specific or generic member name.

ISPF Packing
Provided that the data set is a sequential, PDS or PDSE file and an I/O exit routine is not used, one of these options can be used to control the print behavior when processing data that is in ISPF PACK format.

1. Unpack
If the data set is packed, it is unpacked before printing.

2. None
No checking or processing of ISPF packed data occurs. Packed records are processed and printed in packed format. This option is forced if an I/O exit has been used.

3. Skip
If the input data is packed, no processing or printing occurs.

Additional print options
Displays the Print Options panel allowing you to specify additional print options.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Child panels

- "Print Options panel" on page 619
- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

Related tasks and examples

- "Selecting a range of PDS(E) members" on page 34

Record Identification Criteria panel

The Record Identification Criteria panel is used to set record identification criteria in copybook templates, using the "by field" method (see "Selecting records with templates" on page 203).

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager           Record Identification Criteria              Line 1 of 7

Cmd Con ( Field Name                            Op  Value                      )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
___         REC-TYPE01
___  AND  _ REC-TYPE                                ________________________ _
___  AND  _ NAME                                    ________________________ _
___  AND  _ EMPLOYEE-NO                             ________________________ _
___  AND  _ AGE                                     ________________________ _
___  AND  _ SALARY                                  ________________________ _
___  AND  _ FILLER                                  ________________________ _
*** ****  End of data  ****


Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 212. Record Identification Criteria panel

Cmd Prefix command area - used to enter a template editor prefix command.

Con Connectors - used to specify the AND/OR connectors. These are only

relevant to lines containing record selection expressions. The default value is **AND**. You can overtype the connectors with any of the following values:

Value entered	Resultant value
AND	AND
&	AND
OR	OR
|	OR

Note: Abbreviations are accepted. For example, if you overtype `AND` with `O`, it is changed to `OR`.

(Left Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record selection criteria. The number of left parentheses must match the number of right parentheses.

Field Name
A non-editable field that displays the field names defined in the original copybook.

Use the STR primary command to toggle the displaying of the structure (level information) in the **Field Name** area of the display. The current setting is remembered for future sessions.

Note: If you are running a screen width of 115 or greater, the Field Name field can be displayed at the same time as the Start, Length and Type fields. However, at any width less than this, you need to toggle the display between Field Name and Start, Length and Type. The default display is Field Name. To toggle the display, ensure that your cursor is not in a Value field and press the Right function key (F10) or Left function key (F11).

Start A non-editable field that displays the starting location of the fields, as defined in the original copybook.

Length
A non-editable field that displays the length of the fields, as defined in the original copybook.

Type A non-editable field that displays the data type of the fields, as defined in the original copybook.

Op Operator - used to create record identification expressions. Valid operators are:

=, EQ True if the terms are equal (numerically or when padded)

\=, ¬=, /=, NE
True if the terms are not equal (inverse of =)

>, GT Greater than

<, LT Less than

><, <> Greater than or less than (same as not equal)

>=, GE
Greater than or equal to

\<, ¬< Not less than

<=, LE Less than or equal to

\>, ¬> Not greater than

== True if terms are strictly equal (identical)

\==, ¬==, /==
True if the terms are NOT strictly equal (inverse of ==)

>> Strictly greater than

<< Strictly less than

>>= Strictly greater than or equal to

\<<, ¬<<
Strictly NOT less than

<<= Strictly less than or equal to

ACO Contains all of the values specified. Matching case sensitive. The negative form is XCO.

ACU Contains all of the values specified. Matching not case sensitive. The negative form is XCU.

CO Contains one of the values specified. The negative form is ¬CO.

CU Contains one of the values specified. Matching not case sensitive. The negative form is ¬NC.

NC Numeric contains - a number is one of the values. The negative form is ¬NC.

NN (Same as ¬NU.) Field does not contain a valid numeric for the field type. The positive form is NU.

NR Not in range - field < value1 or field > value2. The positive form is RG.

NU Field contains a valid numeric for the field type:

Field type
Test for being valid numeric

BI, FE, FP, PD, and ZD
Checked against the field definition

Other Recognized as numeric if the display value or character value the field contains is a valid number. The valid number can contain anything that is valid for a floating-point external or decimal number. For example:

```
-1234.55
+3.4621E+10
```

The negative form is ¬NU.

RG Range - value1 >= field <= value2. The negative form is NR.

XCO Does not contain all of the values. Matching case sensitive. The positive form is ACO.

XCU Does not contain all of the values specified. Matching case sensitive. The positive form is ACU.

¬CO Does not contain one of the values specified. The positive form is CO.

¬CU Does not contain any of the values specified. Matching not case sensitive. The positive form is CU.

¬NC Does not contain the specified number(s). The positive form is NC.

¬NU (Same as NN.) Field does not contain a valid numeric for the field type. The positive form is NU.

VER Verify this field is composed only of characters specified in the **Value** column.

Value Use this field to provide a character or numeric constant to be used for comparison in the expression. If the operator supports more than one value (for example, RG, CO) then use comma delimiters to provide separate values. When specifying values, the following rules apply:

Hexadecimal strings
A hexadecimal string must be in the form 'hhhhhh'x . The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters. (0123456789ABCDEF).

Binary strings
A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of 0 and 1s.

Numeric Operands
The value(s) specified for a numeric field type must be a valid number. Quoted values are not accepted.

Character strings
For non-numeric types the value is automatically enclosed in quotes if you do not specify them. When specifying more than one value for operations other than RG and NR you must provide quotes for all the comma delimited values

) Right Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record identification criteria. The number of left parentheses must match the number of right parentheses.

Parent panels

- "Field Selection/Edit panel" on page 565

Related tasks and examples

- "Selecting records with templates" on page 203

Record Sampling panel

The Record Sampling panel is used to specify a *repeating sample* pattern that will limit the number of records presented in an editor session.

In a repeating sample:

1. The **Start position** is set first, and always applies to the physical records, before any template processing is applied.
2. If a template is in use, the template selection criteria then filters the data set and the **Initial skip field** is used to determine the number of selected records to skip.
3. Records are then read from the data set and template processing (if a template is used) occurs on them. Records are included in the Edit session until the

number of selected records (if not using a template, the number of physical records) matches the **Include selected records** field value.

4. After the above number of selected records have been included, the number of selected records in this field will then be skipped. (If not using a template this is physical records) They are not presented in any form in the editor session.
5. Steps 3 and 4 are then repeated until either the end of the data set, or the total number of selected records has been included as specified (in the Total selected record limit field) (if using a template), or the total number of physical records as specified has been read, or File Manager can contain no more records in virtual storage.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                    Record Sampling

Positioning:
   Starting position . . . . _________
   Start position type . . . _ 1. Key
                               2. RBA
                               3. Record number
                               4. Formatted key
Sampling:
   Include physical records  _________
      Skip physical records  _________
Total physical record limit  _________

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 213. Record Sampling panel, when no template is used

Starting position
This represents the physical record number to set the starting point of the editor session. The default is record 1, meaning the top of the data set. A value entered in the **Start position** field on the View Entry or Edit Entry panel, is carried through to this panel when it is opened.

Start position type
Determines how File Manager interprets the value provided in the **Starting position** field. If the **Start position type** is not specified, File Manager assumes that the value is a record number.

Note: The available selections below are changed so that only those valid for the selected file type appear.

1. KEY
Only valid when the data set is a VSAM KSDS, VSAM AIX, or VSAM PATH.

2. RBA
Only valid when the data set is a VSAM KSDS, VSAM AIX, or VSAM ESDS.

3. Record Number
Default. Valid for any type of supported data set.

4. Formatted key
Only valid when the dataset is a VSAM KSDS, VSAM AIX, or VSAM PATH and when a template is provided. This displays the formatted Key Positioning panel.

Note: For templates with more than one layout, a 01 selection list is displayed first.

Include physical records
This represents the number of physical records that is to be included in a repeating sample from the data set. When this field is used, both this field and the **Skip physical records** fields must have nonzero values.

Skip physical records
This represents the number of physical records that is to be skipped in a repeating sample from the data set. The **Include physical records** field must have a nonzero value for this field to have any meaning.

Total physical record limit
This represents the total number of physical records that wil be read from the data set. If the **Include physical records** and **Skip physical records** fields are used, only the included records are counted in this total. The editor session will contain records from the data set up to the end of the data set, or the total number of physical records as specified in this field has been read, or File Manager can contain no more records in virtual storage, whichever occurs first.

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Record Sampling

Positioning:
   Starting position . . . . _________
   Start position type . . . _ 1. Key
                               2. RBA
                               3. Record number
Sampling:
   Initial skip  . . . . . . _________
   Include selected records  _________
      Skip selected records  _________
Total selected record limit  _________

Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 214. Record Sampling panel, when a template is used

Initial skip
This field is only available when processing with a template. This represents the number of records regarded as selected by the template processing that will initially be skipped. These records are not presented in any form in the editor session.

Include selected records
This represents the number of selected records that are included in a repeating sample from the data set. When this field is used, both this field and the **Skip selected records** fields must have nonzero values.

Skip selected records
This represents the number of selected records that are skipped in a repeating sample from the data set. The **Include selected records** field must have a nonzero value for this field to have any meaning.

Parent panels

- "Browse Entry panel" on page 459
- "Edit Entry panel" on page 541
- "View panel" on page 707

Child panels

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Starting and ending editor sessions" on page 51

Record Selection Criteria panel

The Record Selection Criteria panel is used to set record selection criteria in copybook templates, using the "by field" method (see "Selecting records with templates" on page 203).

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager      Record Selection Criteria                    Line 1 of 6

Cmd Con ( Field Name                          Op  Value                      )
    <-> - <---+----1----+----2----+----3----> <-> <---+----1----+----2----+-- -
*** ****  Top of data  ****
          REC-TYPE01
___ AND _ REC-TYPE                            ___ ____________________________ _
___ AND _ NAME                                ___ ____________________________ _
___ AND _ EMPLOYEE-NO                         ___ ____________________________ _
___ AND _ AGE                                 ___ ____________________________ _
___ AND _ SALARY                              ___ ____________________________ _
___ AND _ FILLER                              ___ ____________________________ _
*** ****  End of data  ****

Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 215. Record Selection Criteria panel

Cmd Prefix command area - used to enter a template editor prefix command.

Con Connectors - used to specify the AND/OR connectors. These are only relevant to lines containing record selection expressions. The default value is **AND**. You can overtype the connectors with any of the following values:

Value entered
Resultant value

AND AND

& AND

OR OR

| OR

Note: Abbreviations are accepted. For example, if you overtype AND with O, it is changed to OR.

(Left Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record selection criteria. The number of left parentheses must match the number of right parentheses.

Field Name
A non-editable field that displays the field names defined in the original copybook.

Use the STR primary command to toggle the displaying of the structure (level information) in the **Field Name** area of the display. The current setting is remembered for future sessions.

Note: If you are running a screen width of 115 or greater, the Field Name field can be displayed at the same time as the Start, Length and Type fields. However, at any width less than this, you need to toggle the display between Field Name and Start, Length and Type. The default display is Field Name. To toggle the display, ensure that your cursor is not in a Value field and press the Right function key (F10) or Left function key (F11).

Op Operator - used to create record selection expressions. Valid operators are:

=, EQ True if the terms are equal (numerically or when padded)

\=, ¬=, /=, NE
True if the terms are not equal (inverse of =)

>, GT Greater than

<, LT Less than

><, <> Greater than or less than (same as not equal)

>=, GE
Greater than or equal to

\<, ¬< Not less than

<=, LE Less than or equal to

\>, ¬> Not greater than

== True if terms are strictly equal (identical)

\==, ¬==, /==
True if the terms are NOT strictly equal (inverse of ==)

>> Strictly greater than

<< Strictly less than

>>= Strictly greater than or equal to

\<<, ¬<<
Strictly NOT less than

<<= Strictly less than or equal to

ACO Contains all of the values specified. Matching case sensitive. The negative form is XCO.

ACU Contains all of the values specified. Matching not case sensitive. The negative form is XCU.

CO Contains one of the values specified. The negative form is ¬CO.

CU Contains one of the values specified. Matching not case sensitive. The negative form is ¬NC.

NC Numeric contains - a number is one of the values. The negative form is ¬NC.

NN (Same as ¬NU.) Field does not contain a valid numeric for the field type. The positive form is NU.

NR Not in range - field < value1 or field > value2. The positive form is RG.

NU Field contains a valid numeric for the field type:

Field type
Test for being valid numeric

BI, FE, FP, PD, and ZD
Checked against the field definition

Other Recognized as numeric if the display value or character value the field contains is a valid number. The valid number can contain anything that is valid for a floating-point external or decimal number. For example:

```
-1234.55
+3.4621E+10
```

The negative form is ¬NU.

RG Range - value1 >= field <= value2. The negative form is NR.

XCO Does not contain all of the values. Matching case sensitive. The positive form is ACO.

XCU Does not contain all of the values specified. Matching case sensitive. The positive form is ACU.

¬CO Does not contain one of the values specified. The positive form is CO.

¬CU Does not contain any of the values specified. Matching not case sensitive. The positive form is CU.

¬NC Does not contain the specified number(s). The positive form is NC.

¬NU (Same as NN.) Field does not contain a valid numeric for the field type. The positive form is NU.

VER Verify this field is composed only of characters specified in the **Value** column.

Value Use this field to provide a character or numeric constant to be used for comparison in the expression. If the operator supports more than one value (for example, RG, CO) then use comma delimiters to provide separate values. When specifying values, the following rules apply:

Hexadecimal strings
A hexadecimal string must be in the form 'hhhhhh'x. The value

enclosed in quotes must be an even number of characters and contain valid hexadecimal characters. (0123456789ABCDEF).

Binary strings
A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of 0 and 1s.

Numeric Operands
The value(s) specified for a numeric field type must be a valid number. Quoted values are not accepted.

Character strings
For non-numeric types the value is automatically enclosed in quotes if you do not specify them. When specifying more than one value for operations other than RG and NR you must provide quotes for all the comma delimited values.

) Right Parenthesis - Parentheses must only be used on lines containing expressions. They are used to group record selection criteria. The number of left parentheses must match the number of right parentheses.

Parent panels

- "Field Selection/Edit panel" on page 565

Related tasks and examples

- "Selecting records with templates" on page 203

Record Type Mapping panel

This panel is used to specify which record type, in a copybook or template that contains multiple records types, is to be mapped to the records in a "From" or "Old" template. It is displayed when "edit template" has been requested from the "To" panel of the Copy Utility, the "New" panel of the Compare Data Utility or the Map To panel of the Template Workbench, and the "To" or "New" template or copybook contains multiple record types.

Note: If called from the Compare Data Utility, the panel displays all labels with "New" instead of "To" and "Old" instead of "From".

Panel and field definitions

```
 Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────
File Manager                 Record Type Mapping                  Line 1 of 2

To           USERID.TEMPLATE(FMNCCPY2)
From         USERID.TEMPLATE(COPYTST1)

Cmd To Field Name               Len   From Field Name                    Len
    ****  Top of data  ****
    NEW-TYPE01                   84 | REC-TYPE01                        80
    NEW-TYPE02                  100 |
    ****  End of data  ****









 Command ===> ____________________________________________________ Scroll PAGE
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Describe
  F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 216. The Record Type Mapping panel

TO (or NEW) *template/copybook*
The name of the copybook or template that contains the record descriptions for the "To" or "New" data set.

FROM (or OLD) *template/copybook*
The name of the copybook or template that contains the record descriptions for the "From" or "Old" data set.

(To or New) Field name
Shows the field name and length of the level-01 fields in the "To" or "New" template.

(From or Old) Field name
Shows the field name and length of the level-01 fields in the "From" or "Old" template that have been mapped to the "To" or "New" level-01 fields.

Cmd Prefix command area - used to enter a template editor prefix command. Enter `S` or `E` to edit the field mapping.

Parent panels

Child panels

Related tasks and examples

Record Type Selection panel

The Record Type Selection panel is used when creating or editing templates generated from copybooks that contain more than one record type, or one or more segment types. It is used to select a specific record type for editing or display. It is also used to specify that the record type definitions are to be applied to segmented data records.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager               Record Type Selection                 Line 1 of 2

Processing Option: /  Template for segmented data               CCSID: _____
Cmd SIE Field Name                                      Prompt   Offset Length
        ****  Top of data  ****
___  SI  DEPT                                                        0     25
___  SI  EMPLOYEE                                                    0     30
___      ****  End of data  ****









Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 217. Record Type Selection panel

Template for segmented data
If selected (by typing a "/"), the template can be used to access data sets containing segmented data. The template must contain Record Identification Criteria for each Record Type. Each segment in the data set is treated as a logical record within the function, using the Record Type definition.

Default setting is unselected.

CCSID
The CCSID to be associated with all alphanumeric fields in the template unless the field is in a record which has a CCSID associated with it. If this field is changed those alphanumeric fields in records which do not have an associated CCSID and have the same CCSID will have the CCSID associated with it changed.

Cmd Prefix command area - used to enter a template editor prefix command. You can type prefix commands in multiple Cmd fields and then execute all the commands in one action, by pressing Enter. Line commands available on this panel are:

- **S** Toggles the selection of a segment layout for processing.

  When a segment type is deselected, its field selection criteria are still used to determine whether all segments in the physical record are extracted.
- **E** Display the Field Selection/Edit panel, so that you can edit field selection and expression criteria.

SIE Select/Identification/Expression column.

- **S** Indicates that the record type has been selected for processing.
- **I** Indicates that a record identification expression has been defined for the record type.
- **E** Indicates that a record selection expression has been defined for the record type.

Field name
Shows the field name of the record type (level-01 name).

Prompt
Indicates when the record type has been edited within the current session.

Offset Displays the current offset for each record type. You can overtype the value in this column to change the offset for each record type, or you can enter the OFFSET primary command to change the offset value for one or more record types.

Length
Shows the length of the record. If the record is variable length, this field shows the maximum record length.

Available commands

The following primary commands are available on this panel.

The primary commands that are available on this panel are:

- "DESCRIBE primary command" on page 760
- "FIND/FX primary command" on page 769
- "RESET primary command" on page 806
- "RUNTEMP primary command" on page 812
- "SAVE primary command" on page 812
- "SAVEAS primary command (templates)" on page 813
- "SELECT primary command" on page 814
- "TPRINT primary command" on page 825

Parent panels

Child panels

Related tasks and examples

Redefines / Range Specifications panel

File Manager displays the Redefines / Range Specifications panel when you enter the X line command in the prefix command field (**Cmd**) against a copybook on the Copybook Selection panel.

Panel and field definitions

```
 Process   Options   Help
_ ┌──────────── COPY01 Redefines / Range Specifications ──────────────┐ ──────
F │                                                                   │
  │ Redefines to New Layout:                                          │
C │    Level . . . . . __                                             │
* │    Field name  . . . . ________________________________           │
X │    Set offset  . . . . _        (Enter "/" to select)             │
* │    COBOL level change  _        (Enter "/" to select)             │
  │                                                                   │
  │ Source Range:                                                     │
  │    From statement  . . ________                                   │
  │    To   statement  . . ________                                   │
  │                                                                   │
  │    From string . . . . ________________________________           │
  │    To   string . . . . ________________________________           │
  │ Command ===> ____________________________________________________ │
  │  F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   │
  │  F9=Swap     F12=Cancel                                           │
  └───────────────────────────────────────────────────────────────────┘


Command ===>                                                  Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=LibList   F7=Up
 F8=Down      F9=Swap     F12=Cancel
```

Figure 218. Redefines / Range Specifications panel

Redefines to New Layout

Use the following fields to direct File Manager to generate multiple record layouts for a source that contains many record definitions within one level-1 structure:

Level Source level value used to identify REDEFINES or UNION-level clauses that are to generate new record layouts. This value is used when you have a structure that has defined multiple layouts using COBOL REDEFINES or PL/I unions. File Manager creates a separate record layout for each REDEFINES clause at the given level. After the first matching REDEFINES is found, File Manager creates new layouts for the latter and subsequent REDEFINES clauses for the same level and start location. Header and trailing data items are included in each record layout.

Field name

Specifies the target field name of a REDEFINES clause or the field name with a UNION clause that is used to identify the REDEFINES or UNION statements that are to be used to direct File Manager to create new layouts. This is an alternative way of identifying REDEFINES and UNION statements that require new layouts to be generated. If you specify both **Level** and **Field Name** values, File Manager checks both when generating new layouts.

Set offset
When selected (by typing a "/"), File Manager adjusts the offset value for each layout so that the starting location is the REDEFINES or UNION field start location. Use this option when your record layouts do not include the header data items in the structure. You can only specify this option in conjunction with the previous fields.

COBOL level change
When selected (by typing a "/"), it changes the way the level **Level** value (specified above) is processed. Every occurrence of the specified level value is changed to 01 before the compile step, regardless of whether the data item has a REDEFINES clause or not.

This is a COBOL only option and should be used with care as it can alter a structure from its intended programmable form.

Source Range
Specify the following fields to direct File Manager to extract a subset of the copybook to be compiled into the template:

From statement
Start source line number. This tells File Manager the starting statement for extract. If not specified, the default start location is statement 1.

To statement
End source line number. This tells File Manager the ending statement for extract. If not specified, the default end location is the last source statement.

From string
File Manager searches each source statement for the specified string and the first statement with a matching string starts the extract process. If you specify a "from statement", then the first statement that matches either the **From statement** value or contains the **From string** value begins the extract.

To string
File Manager searches each source statement for the specified string and the first statement with a matching string ends the extract process. If you specify a "to statement", then the first statement that matches either the **To statement** value or contains the **To string** value ends the extract.

Parent panels

"Copybook Selection panel" on page 505

Child panels

Related tasks and examples

- "Advanced copybook selection" on page 147

Related ID expression panel

You use the Related ID expression panel to specify a related ID expression for a layout in a segmented template.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager    Related Id expression for DETAIL-0101             Line 1 of 7

-------- Criteria - Enter 0 to specify related ID expression by field --------
0 Rid: #5='01'                                                               +
Offset ____0          Enter "/" to OR with related ID
Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                 ****  Top of data  ****
___ ___       1  1 HEADER-01                                   AN      1     20
___ ___       2   2 ALLGROUP                                   AN      1     20
___ ___       3    3 TYPE-FIELD                                AN      1      8
___ ___       4     4 TYPECONST                    X(6)       AN      1      6
___ ___       5     4 TYPEREC                      X(2)       AN      7      2
___ ___       6    3 HEADERTEXT                                AN      9     12
___ ___       7     4 HEAD01-TEXT                  X(12)      AN      9     12
___ ___          ****  End of data  ****




Command ===> ____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 219. Related ID expression panel

0 Rid Used to specify the related ID expression.

Offset Offset field - used to enter an offset value. An offset value is a negative or positive integer, between -32760 and +32760, that is added to the record length of the Level 01 field and to the starting position of all fields within the record type, shifting the layout left or right in relation to the records being processed. An offset value of 0 removes a previously supplied offset.

Enter "/" to OR with related ID
By default, identification criteria are ANDed with related ID criteria if both are specified. Select this option to OR the identification criteria with any specified related ID criteria.

Cmd Prefix command area - used to enter a template editor prefix command.

Seq Sequence field - used to order fields for display in File Manager editor sessions and printing data via the File Manager Print utility. The default display/print order is "selected fields in field reference order".

SHE Select/Hold/Create field - shows the field status:

- **S** The field has been selected for processing.
- **H** The field has been selected to be held on the edit display.
- **E** The field has been previously edited to add information such as data create, field headings, and scrambling options.

Ref Field reference - shows the field reference number assigned by File Manager to the field name. Field reference numbers are assigned to all fields in a record, and are used to identify fields in identification or selection criteria expressions.

Key Key segment sequence. Shows the key segment sequence numbers used to

define a multi-segment key field for data set comparison. The display of the Key column can be toggled on or off using the KEYFLDS command.

Field Name
Shows the level number and field name of the field, together with other attributes such as redefines, array sizes and depending on references. If the field is an array, the dimensions are shown in brackets after the field name. If the field has been selected for processing, the field name is highlighted.

Picture
Shows the picture clause for COBOL and for PL/I. Shows the length and scale (if non zero) for binary and packed fields. Also shows the bit length for bit fields.

Type Shows the data type of the field:

AN Alphanumeric (includes alphabetic, and group items)

AX Alphanumeric displayed in long hexadecimal.

BI Binary

BT Bit

DB DBCS

FE Floating point (external)

FP Floating point (internal)

G Graphic

PD Packed decimal (internal decimal)

VC Variable Character

VD Variable DBCS

VG Variable Graphic

ZC Character Null terminated

ZD Zoned decimal (external decimal)

ZE Zoned decimal edited

ZG Graphic Null terminated

Z2 DBCS Null terminated

Numeric data types supported by File Manager include numeric-edited data types.

Start Shows the start column of the field. For variable located fields the start location is based on the maximum length of the record.

Length
Shows the length of the field. For a level-01 field, if the record is variable length, the length shown is the maximum record length.

Available commands

The prefix commands that can be entered in the Cmd field are:

E Displays the Field Attributes panel for this field. You can use the Field Attributes panel to specify the following:

- For a numeric field, whether you want leading zero suppression when the field is displayed on the screen or printed using the SNGL or TABL display or print formats.
- When you use the template to help generate test data, what data pattern File Manager should use for the field.

H Set Hold status for a single field, if the field is already in Hold status the Hold status is removed. When a field has the Hold status set it remains in the editor window during left/right scroll operations.

Hn Set Hold status for n fields, if any of the fields are already in Hold status, the Hold status is removed.

HH Select a block of fields to have the Hold status set. If any of the fields are already in Hold status, the Hold status is removed. Use the HH prefix command to mark the start and end of the block of fields.

S Select a single field for processing or, if the field is already selected for processing, deselect it. If you select multiple fields for processing using the S prefix command, the fields are displayed in the order they appear in the template.

Sn Select *n* fields for processing or, if any of the fields are already selected for processing, deselect them.

S* Select all fields from the current field to the end of the list for processing or, if any of the fields are already selected for processing, deselect them. If you want to exclude just a few fields, you can use S* to explicitly select all the fields in a fresh template, then use S to deselect the fields you want to exclude.

SS Select a block of fields for processing or, if any of the fields are already selected for processing, deselect them. Use the SS prefix command to mark the start and end of the block of fields.

X Toggle between a display type of AN (character) and AX (long hexadecimal). This command has no effect on non-alphanumeric fields.

X*n* Perform the X command against all fields from the current field for *n* fields.

X* Perform the X command against all fields from the current field to the end of the list.

XX Perform the X command against a block of fields. Use the XX prefix command to mark the start and end of the block of fields.

The primary commands that are available on this panel are:

- "DESCRIBE primary command" on page 760
- "FIND/FX primary command" on page 769
- "RESET primary command" on page 806
- "RUNTEMP primary command" on page 812
- "SAVE primary command" on page 812
- "SAVEAS primary command (templates)" on page 813
- "SELECT primary command" on page 814
- "TPRINT primary command" on page 825

Parent panels

- "RID Selection panel"

Child panels

Related tasks and examples

- "Editing related ID criteria" on page 167

RID Selection panel

After you have selected a layout within a template for segmented data for which you want to specify related ID criteria, the RID Selection panel lists all the other level-01 layouts in the template. You then use the RID Selection panel to select a specific layout to use for specifying ID criteria.

Panel and field definitions

```
 Process   Options   Help
_____________________________________________________________________________
File Manager           RID Selection for DETAIL-0101              Line 1 of 8

E against layouts to specify related id criteria
Cmd SIE Field Name                                        Prompt   Offset Length
        ****  Top of data  ****
___ S   HEADER-01                                                      0     20
___ S   HEADER-02                                                      0     20
___ S   HEADER-03                                                      0     20
___ S   DETAIL-0102                                                    0     30
___ S   DETAIL-0103                                                    0     30
___ S   DETAIL-0201                                                    0     30
___ S   DETAIL-0202                                                    0     30
___ S   DETAIL-0203                                                    0     30
___     ****  End of data  ****









Command ===> _____________________________________________________ Scroll CSR
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=RunTemp
 F7=Up        F8=Down      F9=Swap     F12=Cancel
```

Figure 220. RID Selection panel

Cmd Prefix command area - used to enter a template editor prefix command. You can type prefix commands in multiple Cmd fields and then execute all the commands in one action, by pressing Enter. Line commands available on this panel are:

S Toggles the selection of a segment layout for processing.

When a segment type is deselected, its field selection criteria are still used to determine whether all segments in the physical record are extracted.

E Display the Related ID expression panel, so that you can edit related ID expression criteria.

SIE Select/Identification/Expression column.

S Indicates that the layout has been selected for processing.

I Indicates that the layout has an identification expression.

E Indicates that the layout has a selection expression.

Field name
Shows the field name of the related ID (level-01 name).

Prompt
Indicates when the related ID has been edited within the current session.

Offset Displays the current offset for each related ID. You can overtype the value in this column to change the offset for each related ID, or you can enter the OFFSET primary command to change the offset value for one or more related IDs.

Length
Shows the length of the related ID. If the related ID is variable length, this field shows the maximum length.

Available commands

The primary commands that are available on this panel are:

- "DESCRIBE primary command" on page 760
- "FIND/FX primary command" on page 769
- "RESET primary command" on page 806
- "RUNTEMP primary command" on page 812
- "SAVE primary command" on page 812
- "SAVEAS primary command (templates)" on page 813
- "SELECT primary command" on page 814
- "TPRINT primary command" on page 825

Parent panels

- "Field Selection/Edit panel" on page 565

Child panels

- "Related ID expression panel" on page 639

Related tasks and examples

- "Editing related ID criteria" on page 167

Replace panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                        Replace

To Data set:
   Data set/path name  . . 'FMN.FMDATA'
   Member name (or mask) . ________
   Volume serial . . . . . ______

Allocation Options:
1  1. Allocate using the attributes of
        'FMNUSER.DATA'
   2. Specify allocation attributes

Processing Options:
ISPF Packing                         Enter anything to select option
1  1. None                           _  Binary mode, reclen _____
   2. Pack



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 221. Replace panel

The **Allocation Options** are only required when the "To" data set does not exist.

The **Binary mode** and **reclen** parameters are only relevant for an HFS file.

Parent panels

Child panels

- “Allocate panel” on page 448
- “Allocate (Two) panel” on page 451
- “Allocate (Three) panel” on page 453

Equivalent functions

- None.

Related tasks and examples

- “Creating, replacing, and adding data using existing data” on page 246

Saveas panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                        Saveas

To Data set:
   Data set/path name  . . 'FMN.FMDATA'
   Member name (or mask) . ________
   Volume serial . . . . . ______

Allocation Options:
1  1. Allocate using the attributes of
        'FMNUSER.DATA'
   2. Specify allocation attributes

Processing Options:
ISPF Packing                         Enter anything to select option
1  1. None                           _  Binary mode, reclen _____
   2. Pack



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 222. Saveas panel

The **Allocation Options** are only required when the "To" data set does not exist.

The **Binary mode** and **reclen** parameters are only relevant for an HFS file.

Parent panels

Child panels

- "Allocate panel" on page 448
- "Allocate (Two) panel" on page 451
- "Allocate (Three) panel" on page 453

Equivalent functions

- None.

Related tasks and examples

- "Creating, replacing, and adding data using existing data" on page 246

Scramble Exit Specification panel

The Scramble Exit Specification panel allows you to to provide the name of a scramble exit, a constant to be passed to the exit, and to specify whether numeric fields are to be formatted before the value is passed to the exit.

Panel and field definitions

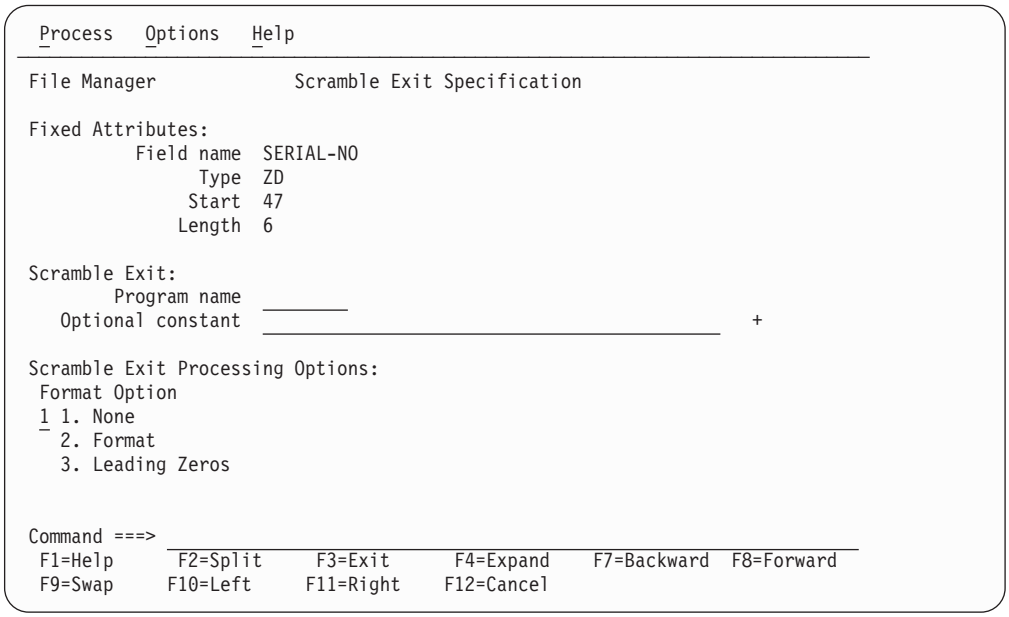


Figure 223. Scramble Exit Specification panel

Program name
A 1 to 8 character valid load module member name that identifies the scramble exit invoked during a copy operation for this output field. The exit must be in the form of a load module, in any load library available to File Manager at the time of the copy process, either by a STEPLIB DD statement, or in LINKLIST, or LPALIST.

Optional constant
A constant value that is passed to the exit for each call type. To enter a hexadecimal value, press F4 or enter the EXPAND command with the cursor positioned in this field and use the command HEX ON from the expand window.

Format Option
This option is only displayed for numeric fields and allows you to request File Manager to pass the value specified in **Optional constant** to the exit in one of the following forms:

1. None
Field value passed to the exit in the internal format.

2. Format
Field value passed to the exit as displayed numeric with leading zeros suppressed.

3. Leading Zeros
Field value passed to the exit as displayed numeric with leading zeros.

Parent panels

- "Field Attributes panel - alphanumeric fields" on page 553
- "Field Attributes panel - numeric field" on page 559

Related tasks and examples

- "Scrambling data" on page 262

Sequential Data to Tape panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager              Sequential Data to Tape

Tapes:  No tapes allocated


Input:
   Data set name . ____________________________________________
   Member  . . . . ________      if partitioned
   Volume serial . ______        if not cataloged
   Unit  . . . . . ________      for tape data sets
   Skip count  . . ________      number of records to be skipped
   Punch count . . ALL_____      number of cards to be punched

Output:
   DDNAME to use . ________      enter new name, or select one from above
   Tape mode . . . __            optional recording mode or density code
   Record format . ___           F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Block size  . . _______       required for blocked output

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 224. Sequential Data to Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Set Batch Job Card Information panel (option 0.4)

Panel and field definitions

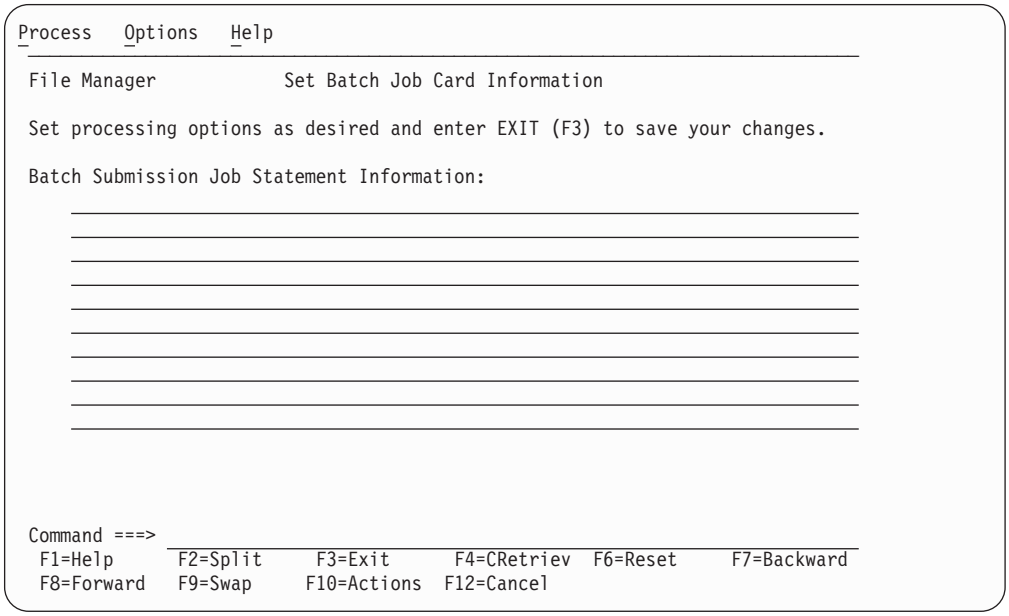


Figure 225. Set Batch Job Card Information

Batch Submission Job Statement Information
Batch job card information to be used for generating batch job submission JCL.

If any of these lines are non-blank, then all the non-blank lines are assumed to define a JCL job card and are copied as is into generated JCL. If all these lines are blank, then the JCL generation process generates a basic job card using information from system variables.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- "SET (Set Processing Options)" on page 1047

Related tasks and examples

- "Setting your default processing options" on page 41

Set COBOL Processing Options panel (option 0.5.2)

The COBOL Processing Options control various options used by the COBOL compiler to generate templates from your COBOL copybooks.

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager            Set COBOL Processing Options

Additional SYSLIB data sets:
1.  ____________________________________________
2.  ____________________________________________
3.  ____________________________________________
4.  ____________________________________________
5.  ____________________________________________
6.  ____________________________________________
7.  ____________________________________________
8.  ____________________________________________
9.  ____________________________________________
10  ____________________________________________

COBOL Replacing Options:
   From string                      To string
1. ________________________ by_____________________________
2. ________________________ by_____________________________
3. ________________________ by_____________________________
4. ________________________ by_____________________________
5. ________________________ by_____________________________

COBOL Compiler Options:
   Enter "/" to select option
   _ DBCS                          _ Decimal-point is comma
   _ Arith(extend)                 _ Mixed case field names
 Additional options ____________________________________________________________

   Maximum Return Code to be accepted from compiler  04  (0-99)
Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap      F10=Actions  F12=Cancel
```

Figure 226. Set COBOL Processing Options

Additional SYSLIB data sets
A concatenation of up to 10 SYSLIB data sets may be entered, these data sets are searched in the order specified for COPY members for the compilation.

Note: CA-Panvalet libraries cannot be specified as additional SYSLIB sets. Further, if a CA-Panvalet Library is the main copybook library, then no additional SYSLIB data sets may be specified.

COBOL Replacing Options
"From" and "To" pseudo-text character strings for the REPLACE compiler-directing statement.

COBOL Compiler Options
Compiler options to be used when compiling a COBOL copybook into a template:

DBCS Instructs File Manager to use the DBCS compiler option.

Decimal-point is comma
Uses the "Decimal-point is comma" SPECIAL-NAMES paragraph when compiling COBOL copybooks.

Arith(extend)
Uses the Arith(extend) COBOL compile option when compiling COBOL copybooks.

Mixed case field names
Retains the original case of the field name as coded in the COBOL copybook.

Additional options
Additional COBOL compiler options which are added via the CBL statement when a COBOL compile is run to build or update a template. These options are validated during the compile process. To avoid compile errors please ensure the syntax is correct and that any additional data sets required by these options are allocated prior to invoking File Manager.

Maximum Return Code
Sets the maximum warning/error code accepted from the compiler, when compiling a copybook. Any code higher than specified causes File Manager to display a pop-up panel, in which you can review the compilation and decide about future processing.

Parent panels

- "Set Processing Options panel" on page 661
- "Set Language and Compiler Specifications (option 0.5)" on page 654

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Printing from File Manager" on page 299
- "Data description support" on page 134
- "Setting your COBOL processing options" on page 142

Set DBCS Format panel

The Set DBCS Format option is used to define where DBCS format, mixed format, and EBCDIC format fields are within an input record.

Panel and field definitions

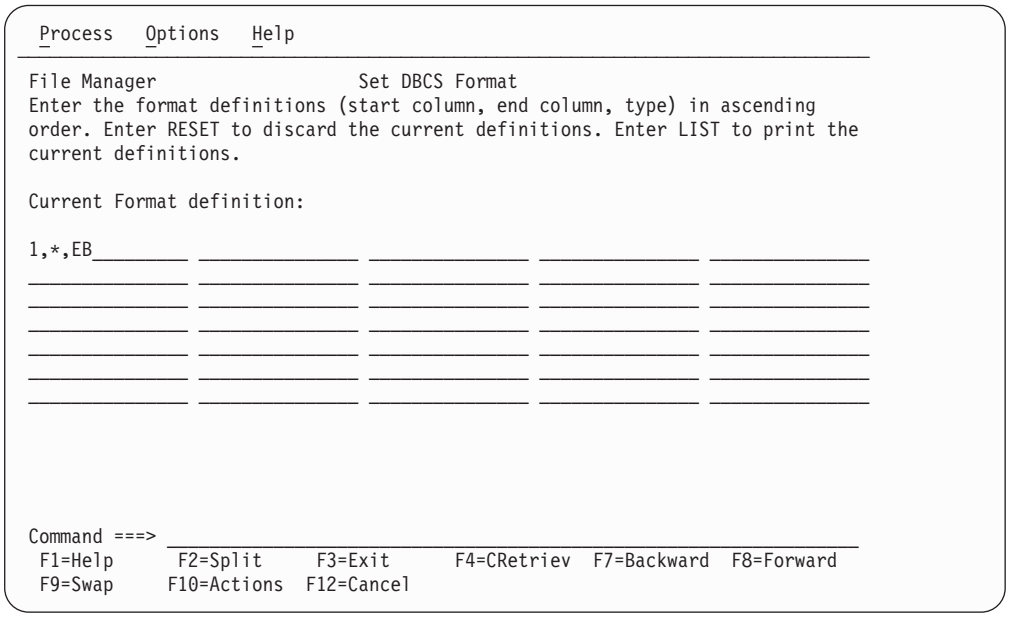


Figure 227. Set DBCS Format panel

Format definition
Defines the start, end, and type of input fields in this order separated by commas. Up to 32 field definitions may be entered in ascending order.

Valid ranges are:

start 1-9999

end 1-9999 or * for end of record

type EB = EBCDIC

DB = DBCS

MI = mixed

Parent panels

- "Utility Functions menu panel" on page 703

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Printing DBCS data" on page 305

Set HLASM Processing Options panel (option 0.5.3)

The HLASM Processing Options panel controls various options used by the HLASM compiler to generate templates from your HLASM copybooks.

Panel and field definitions

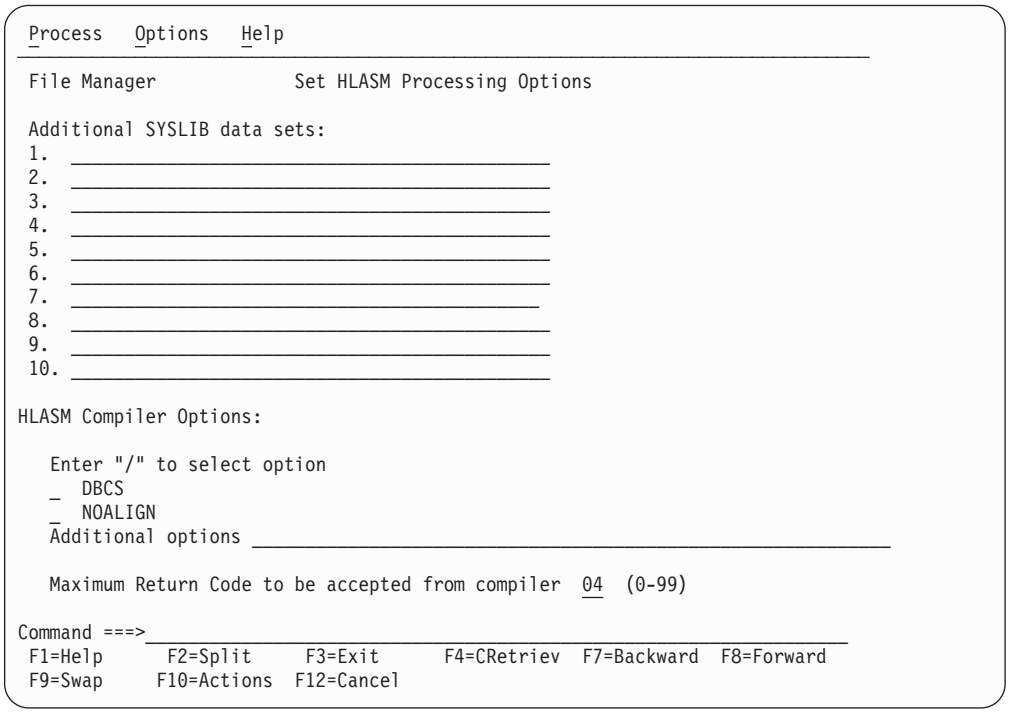


Figure 228. Set HLASM Processing Options

Additional SYSLIB data sets
A concatenation of up to 10 SYSLIB data sets may be entered, these data sets are searched in the order specified for COPY or INCLUDE members for the compilation.

HLASM Compiler Options for Copybooks
Compiler options to be used when compiling a HLASM copybook into a template:

DBCS If selected, use the DBCS compiler option.

NOALIGN
If selected, use the NOALIGN compiler option.

Additional options
Additional HLASM compiler options which are added via the *PROCESS statement when the assembler is run to build or update a template. These options are validated during the assembly process. To avoid assembly errors please ensure the syntax is correct.

Parent panels

- "Set Processing Options panel" on page 661
- "Set Language and Compiler Specifications (option 0.5)" on page 654

Child panels

- None.

Equivalent functions

- "Printing from File Manager" on page 299
- "Setting your HLASM processing options" on page 144

Set Language and Compiler Specifications (option 0.5)

The Language and Compiler Specifications menu panel is displayed when option 5 is selected from the Settings menu.

The Compiler panels allow you to view and modify selected File Manager parameters relating to the compiler to be used when generating templates together with compiler options to be specified.

Panel and field definitions

```
Process   Options   Help
 File Manager            Set language and compiler specifications

 1  LANG      Compiler language selection
 2  COBOL     COBOL compiler specifications
 3  HLASM     HLASM compiler specifications
 4  PL/I      PL/I compiler specifications

Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 229. Set Language and Compiler Specifications

The following topics are presented in sequence by pressing **Enter** or may be selected by number:

1 LANG
Displays the compiler language selection panel.

2 COBOL
Displays the COBOL compiler specifications panel.

3 HLASM
Displays the HLASM compiler specifications panel.

4 PL/I Displays the PL/I compiler specifications panel.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- "Compiler Language Selection panel" on page 470
- "Set COBOL Processing Options panel (option 0.5.2)" on page 649
- "Set PL/I Processing Options panel (option 0.5.4)" on page 656
- "Set HLASM Processing Options panel (option 0.5.3)" on page 652

Equivalent functions

- None.

Set Output Data Set Allocation Options panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager        Set Output Data Set Allocation Options

Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Data set  Space  Primary   Secondary Management Storage   Data      Device
          Units  Quantity  Quantity   Class     Class     Class     Type

Audit Log CYL    16        8         _________ ________  ________  SYSALLDA

Print     CYL    64        32        _________ ________  ________  SYSALLDA









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 230. Set Output Data Set Allocation Options panel

Space Units
Defines the unit of primary and secondary space to be allocated.

BLK Block of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

Primary Quantity
Amount of DASD space to be used for primary space allocation. The range depends on the space unit specified and the DASD device type.

Secondary Quantity
Amount of DASD space to be used for secondary space allocation. The range depends on the space unit specified and the DASD device type.

Management Class
Name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage Class
Name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Data Class
Name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Device Type
Specifies the device type of the volume to contain the data set. Enter an IBM device type or a generic device name that is supported by your system.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- None.

Set PL/I Processing Options panel (option 0.5.4)

The PL/I Processing Options panel controls various options used by the PL/I compiler to generate templates from your PL/I copybooks.

Panel and field definitions

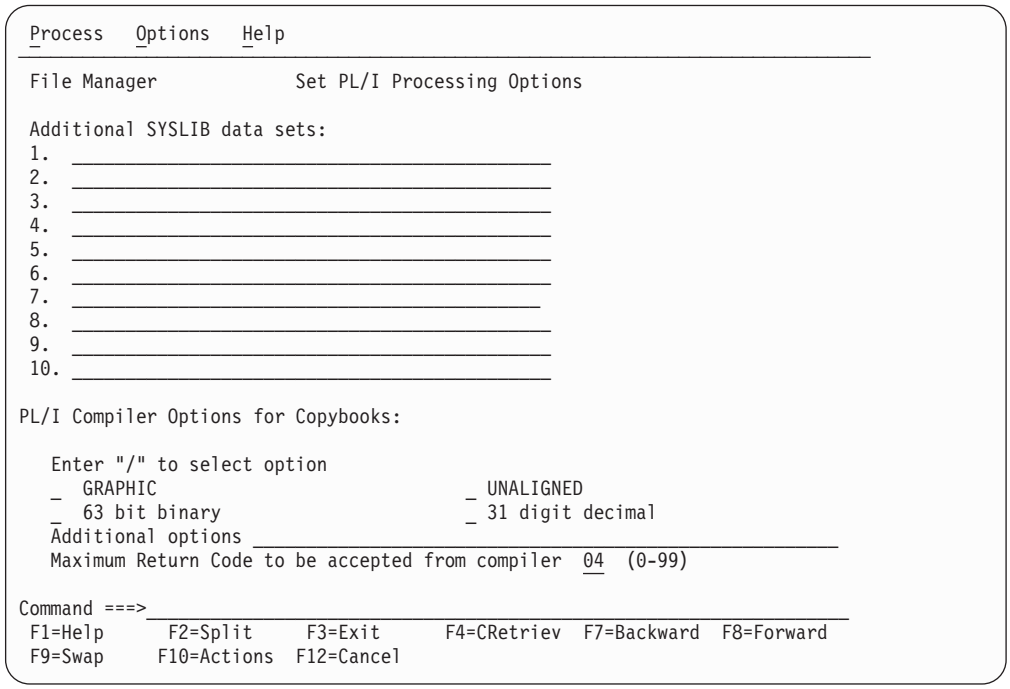


Figure 231. Set PL/I Processing Options

Additional SYSLIB data sets
A concatenation of up to 10 SYSLIB data sets may be entered, these data sets are searched in the order specified for COPY members for the compilation.

PL/I Compiler Options for Copybooks
Compiler options to be used when compiling a PL/I copybook into a template:

GRAPHIC
If selected, use the GRAPHIC compiler option. Otherwise, use NOGRAPHIC.

63 bit binary
If selected, use the LIMITS(FIXEDBIN(63)) compiler option. Otherwise, use LIMITS(FIXEDBIN(31)).

UNALIGNED
If selected, use the DEFAULT RANGE(*) UNALIGNED; language statement to change the default alignment.

31 digit decimal
If selected, use the LIMITS(FIXEDDEC(31)) compiler option. Otherwise, use LIMITS(FIXEDDEC(15)).

Additional options
Additional PL/I compiler options which are added via the *PROCESS statement when a PL/I compile is run to build or update template. These options are validated during the compile process. To avoid compile errors please ensure the syntax is correct and that any additional data sets required by these options are allocated prior to invoking File Manager.

Maximum Return Code
Sets the maximum warning/error code accepted from the compiler, when compiling a copybook. Any code higher than specified causes File Manager to display a pop-up panel, in which you can review the compilation and decide about future processing.

Parent panels

- "Set Processing Options panel" on page 661
- "Set Language and Compiler Specifications (option 0.5)" on page 654

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Printing from File Manager" on page 299
- "Setting your PL/I processing options" on page 144
- "Data description support" on page 134

Set Print Processing Options panel (option 0.1)

The Set Print Processing Options panel is used to set the general printing options for File Manager.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager             Set Print Processing Options

 Print Options:
    Output destination      Record length      Enter "/" to select option
    1  1. SYSPRINT          2  1. 80              Page skip
       2. Terminal             2. 132          /  Wide print
       3. Data set                             /  Translate non-printable chars
       4. REXX                                    Uppercase message text
                                               /  Data prefix
    Data set DISP           Dump format        /  Header page
    2  1. OLD               1  1. Updown
       2. MOD                  2. Across

    Data set name  . . JOHNLEV.FMN.LIST
    Output class . . . A
    Lines per page . . 50
    Record limits  . . (1,*)              (n,m) n=begin column, m=end column


 Command ===> ___________________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
  F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 232. Set Print Processing Options

Output destination
Specifies where print output is sent, except for batch execution.
SYSPRINT
Send print output to the current SYSPRINT allocation.

Terminal
Send print output to the TSO terminal.

Data set
Output is accumulated in the print data set specified in the **Data set name** field. This data set can be browsed using the PB command, or sent for printing to the JES spool queue with class specified in the **Output class** field using the PRINT command while browsing.

REXX Send output to a REXX stem variable. Each line corresponds to a variable named FILEM.*nnn*. FILEM.0 contains the line count.

When you use the PRINT primary command in Print Browse (option 3.9), or if you specify batch execution in a File Manager panel, then print output is always sent to a SYSOUT class.

Record length
Specifies the line length of print output.

80 The line length of print output is 80 characters, suitable for a terminal.

132 The line length of print output is 132 characters, suitable for a printer.

Note: Record length is not applicable when:

- The data is printed in TABL format, or
- You specify the wide format for the compare output listing

In either of the above cases, when data is printed the line length is determined from the number and size of fields printed. If this length is greater than the record length specified for the data set, the print line is truncated. The maximum print line length is 32760.

Data set DISP
Specifies the disposition of a print data set.

OLD The print data set is cleared before each print operation, and print output is written from the beginning of the data set.

MOD Default setting. Print output is appended to the existing print data set.

Dump format
Specifies the format of hexadecimal print output (for example, when you use Tape Print with **Print format** set to HEX). For an example of each format, see "Selecting a display format" on page 66.

Updown
The two digits making up the hexadecimal representation of each EBCDIC character are displayed vertically directly under that character.

Across
The hexadecimal digits are displayed as 2 groups of 4 full words resulting in 32 hexadecimal digits followed by the EBCDIC character display to the right of the hexadecimal display.

Page skip
If selected, print output from each function begins on a new page.

Wide print
If selected, for to-be-allocated (new) print output datasets (online) or for

SYSPRINT allocation (in batch), File Manager uses the maximum record length/blocksize of 32756/32760 and record format of VBA.

If not selected, for to-be-allocated (new) print output datasets (online) or for SYSPRINT allocation (in batch), File Manager uses the record length/blocksize of 133/137 and record format of VBA.

Note: This setting only applies when the output dataset is newly-allocated during execution. It is ignored for pre-allocated print output datasets.

Translate non-printable chars
Specifies how File Manager should translate non-printable characters.

If selected, non-printable characters are translated to blanks using a translate table.

If not selected, no translation is performed. Use PRTTRANS=OFF to support special print chains. This may make printing faster.

For instructions on altering the print translate table, see the *File Manager Customization Guide*.

Uppercase message text
All message text is translated to uppercase.

Data prefix
If selected, when data is printed in CHAR print format, record number and length are included in character-format print output.

If not selected, you can specify DATAHDR=NO to left-justify the data in the print output, without header information.

Header page
If selected, a header page (a title page with File Manager on it) is included in the print output.

If you specified the **Output destination** as **2** (Terminal), then the header page is not included in the print output, even if you have selected this option.

Data set name
If you specified the **Output destination** as **3** (Data set), then after you enter the PRINT command while executing the Print Browse (PB) function, File Manager sends print output to this temporary data set.

The default is *userid*.FMN.LIST, unless changed in your File Manager installation.

Output class
The class of the JES spool queue to be used when the PRINT command is issued while browsing the temporary print data set.

Lines per page
Specifies the number of lines printed on each page of print output. You can specify a value from 1 to 999. The default is 60.

Record limits
When data is printed in CHAR or LHEX print format, limits print output for each record (or OAM object).

(1,*) The entire record (or block) is printed.

(*n*,*m*) Print output is limited to the data from columns (bytes) *n* through *m*. An asterisk (*) specified for *m* indicates the end of the record.

Parent panels

- "Set Processing Options panel"

Child panels

- None.

Equivalent functions

- "SET (Set Processing Options)" on page 1047

Related tasks and examples

- "Printing from File Manager" on page 299
- Printing

Set Processing Options panel

Lists the processing options available within File Manager. Selecting an option displays the relevant Set Processing Options panel.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Set Processing Options

1  Print     Print settings
2  System    System settings
3  Tape      Tape processing options
4  Batch     Job card specifications
5  Compiler  Language and compiler specifications
6  EDIT      Editor options
7  Sharing   VSAM Edit sharing options
8  Temporary Temporary Data Set Allocations
9  Output    Output Data Set Allocations
10 Trace     Trace options




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap      F10=Actions  F12=Cancel
```

Figure 233. Set Processing Options Menu panel

1 Print
Displays the Set Print Processing Options panel

2 System
Displays the Set System Processing Options panel

3 Tape Displays the Set Tape Processing Options panel

4 Batch
Displays the Batch Job Card Information panel

5 Compiler
Displays the Language and compiler specifications panel

6 EDIT
Displays the Editor Options panel

7 Sharing
Displays the VSAM Edit Sharing Options panel

8 Temporary
Displays the Set Temporary Data Set Allocation Options panel (option 0.8)

9 Output
Displays the Set Output Data Set Allocation Options panel

10 Trace
Displays the Set Trace Options panel

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

- "Set Print Processing Options panel (option 0.1)" on page 658
- "Set System Processing Options panel (option 0.2)" on page 663
- "Set Tape Processing Options panel (option 0.3)" on page 665
- "Set Batch Job Card Information panel (option 0.4)" on page 649
- "Set Language and Compiler Specifications (option 0.5)" on page 654
- "Editor Options panel" on page 546
- "VSAM Edit Sharing Options panel" on page 721
- "Set Output Data Set Allocation Options panel" on page 655
- "Set Temporary Data Set Allocation Options panel" on page 667
-

Equivalent functions

- "SET (Set Processing Options)" on page 1047

Related tasks and examples

- "Setting your default processing options" on page 41

Set System Processing Options panel (option 0.2)

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager            Set System Processing Options
Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Data Presentation and Conversion Options:
   PAD  . . OFF               OFF, ON, character or hex value
   NOTRUNC  ___               No truncation if PAD selected
   CYLHD  . ABSOLUTE          ABSOLUTE or RELATIVE
   EOD  . . ________________  Tape input delimiter, 1-8 bytes
   CCSID  . 00037             Default CCSID

Editor choice for viewing output data, members:
   Use File Manager editor _

User I/O Exit Specification:
   Exit enabled  NO           Disabled by installation options
   Default exit               N/A




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 234. Set System Processing Options panel

PAD Specifies whether records are padded or truncated while being copied.

OFF Records are not padded with blanks or a specified character, except where required by the output record format. If the output records are fixed format, they are padded with nulls ('00'x), as required to match the output record length. Variable and undefined format outputs are not padded.

Output records are always truncated if they exceed the maximum record length (for fixed or variable) or maximum block length (for undefined formats). However, trailing blanks that fit within the maximum length are not stripped from the record.

ON Records are modified, depending on the output record format. For fixed-format output, input records that do not fit are truncated or padded with blanks. For variable or undefined output, trailing blanks are stripped from the input record. (For reasonable results, do not try to deblock and pad in one run.)

c The value used for padding records. This can be a single character or a hexadecimal value. For example, you can enter an X followed by two hexadecimal characters enclosed in quotation marks (X'00'). In addition, for variable or undefined output, a trailing series of the specified character is stripped from the input record. Essentially, this option has the same effect as PAD=ON except that the specified character *c* is used for padding or truncation instead of blanks.

To specify a lowercase pad character, for example “i”, either use c'i' or the hexadecimal characters x'89'.

NOTRUNC
Specifies that, if the PAD option has been selected, no truncation is performed when copying or writing records to a variable-length data set.
NO Truncation performed.
YES No truncation performed.

CYLHD
Defines the way you specify the location of a disk data set in disk functions.
ABSOLUTE
File Manager interprets cylinder-head and track-number values as actual physical addresses.
RELATIVE
File Manager interprets cylinder-head and track-number values as relative to the start of the data set.

EOD End-of-data delimiter for tape input.

delimiter
A 1- to 8-character delimiter. For character data, enclose the string in quotation marks if it contains blanks, commas, or lowercase letters. For hexadecimal data, enter an X followed by the string enclosed in quotation marks (for example, X'04').

For some Tape Specific Functions (option 4), you specify the number of files to process in a **Files** field. If you specify EOD in the **Files** field, then files are processed until the first record that begins with the EOD delimiter. However, there is no default EOD delimiter for tape input. So, if you want to specify EOD in the **Files** field, then you must first specify an EOD delimiter on this options panel; otherwise, File Manager issues an error.

CCSID
Specifies the CCSID to be used as the default CCSID when File Manager is run in batch.

Use File Manager editor
This option controls which editor is invoked when viewing or browsing a member from a member selection panel or when viewing generated output. Select this option to invoke the File Manager editor. If the option is not selected, the ISPF Browse, View or Edit facility is invoked when available.

Exit enabled
This field shows whether or not the installation options allow for a user override of the File Manager I/O exit options in the editor panels, or when using the batch or online panels for these functions: Data Set Copy (DSC), Data Set Compare (DSM), Data Set Edit in Batch (DSEB), Data Set Generate (DSG), Data Set Print (DSP), Data Set Update (DSU) and Find/Change Utility (FCH). If the USEIOX installation option was set to DISABLE, this field is protected and cannot be changed by a user. If the USEIOX installation option was set to ENABLE, the field can be edited and users can choose whether or not to allow user I/O exit overrides within their invocation of File Manager.

NO - Disabled by installation options
The USEIOX installation option was set to DISABLE, therefore, you cannot specify a user I/O exit override.

NO - YES or NO
The USEIOX installation option was set to ENABLE, therefore, you

are able to specify a user I/O exit override. However, by setting the field to NO, you are choosing not to allow this in your invocation of File Manager.

YES The USEIOX installation option was set to ENABLE, and you are choosing to allow a user I/O exit override in your invocation of File Manager.

Default exit
When the USEIOX installation option is enabled, you can use this field to specify the file that contains the default exit program. If the USEIOX installation option is set to DISABLE, this field is protected and cannot be changed.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- "SET (Set Processing Options)" on page 1047

Related tasks and examples

- "Setting your default processing options" on page 41
- "Specifying a data set and a member name" on page 16

Set Tape Processing Options panel (option 0.3)

Panel and field definitions

```
Process   Options   Help
--------------------------------------------------------------------------------
File Manager            Set Tape Processing Options
Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Tape Processing Options:
   TAPELBL  SL_                SL, AL or AL4, tape label option
   ASCII  . NO_                NO, IN (YES), OUT, or BOTH









Command ===>_____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

Figure 235. Set Tape Processing Options

TAPELBL
Specifies the type of labels that are created:

AL ANSI Version 3 labels are created. ("Version 3" refers to ANSI X3.27–1978, ISO 1001–1979, and FIPS 79 standards.)

AL4 ANSI Version 4 labels are created. ("Version 4" refers to ANSI X3.27–1987 level 4 and ISO 1001–1986(E) standards.)

SL EBCDIC labels are created.

For detailed information about ANSI tape labels, see *z/OS DFSMS: Using Magnetic Tapes*.

This parameter only affects Initialize Tape (option 4.12).

ASCII Specifies requirements for translation of tape data between ASCII and EBCDIC.

BOTH Translates tape input from ASCII format to EBCDIC format, and translates tape output from EBCDIC format to ASCII format.

IN Translates tape input from ASCII format to EBCDIC format.

NO Does not translate data. This is the default.

OUT Translates tape output from EBCDIC format to ASCII format.

This processing option applies to all tape input and output functions except:

- Initialize Tape
- Tape Browse
- Tape Label Display
- Tape to Labeled Tape
- Tape to Tape Compare
- Tape Update

For Tape Label Display, ASCII translation is automatically performed when needed. For Initialize Tape and Tape Browse, you can specify ASCII input or output when invoking the function. For the other functions, translation is not supported by File Manager.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- "SET (Set Processing Options)" on page 1047

Related tasks and examples

- "Setting your default processing options" on page 41

Set Temporary Data Set Allocation Options panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager      Set Temporary Data Set Allocation Options

Set processing options as desired and enter EXIT (F3) to save your changes.
Enter RESET to restore installation defaults.

Temporary data set allocation defaults:
   Unit . . . . . . . . SYSALLDA                leave blank for default
   Data class . . . . . ________                leave blank for default
   Storage class  . . . ________                leave blank for default
   Management class . . ________                leave blank for default

Default High Level Qualifiers:
   Temporary Data Sets  ____________________    HLQ (opt. &USER/&PREFIX)

Auxiliary Data Set Allocation Defaults:
   High Level Qualifier ____________________    leave blank for default
   Data class . . . . . ________                leave blank for default
   Storage class  . . . ________                leave blank for default
   Management class . . ________                leave blank for default
   Model Data Set Name  ________

   ____________________________________________________
   Auxiliary Data Set Name

   ____________________________________________________
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 236. Set Temporary Data Set Allocation Options panel

Temporary data set allocation defaults

Unit Can be one of these:

- A 3-digit hexadecimal device number.
- A 4-digit hexadecimal device number prefixed with a slash (/).
- Device type (generic name).
- Group name (installation-defined symbolic name).

Data class

Name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage class

Name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Management class

Name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Default High Level Qualifiers

Temporary Data Sets

Can be any multi-level qualifier, which is used by File Manager as the high-level qualifier (HLQ) in names of created datasets. You can use these symbols as part of the HLQ:

&USER

Represents the user ID.

&PREFIX

Represents the TSO prefix.

The default is none (unless changed in your File Manager installation).

Auxiliary Data Set Allocation Defaults

High Level Qualifier

Can be any multi-level qualifier, which is used by File Manager as the high-level qualifier (HLQ) in names of created datasets. You can use these symbols as part of the HLQ:

&USER

Represents the user ID.

&PREFIX

Represents the TSO prefix.

The default is none (unless changed in your File Manager installation).

Data class

Name of a data class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Storage class

Name of a storage class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Management class

Name of a management class defined in your Storage Management System (SMS) installation. Do not enter a value on a system without SMS.

Model Data Set Name

A data set name to be used as the model file for defining the auxiliary data set for a full function edit session. The TSO prefix (as defined for your user ID) is used as the high-level qualifier for any name that is not enclosed in quotes.

A model file should be used to determine the volume placement of the auxiliary data set in a non-SMS environment. The model file must be a VSAM cluster.

Auxiliary Data Set Name

A data set name to be used as the auxiliary file for a full function edit session. The TSO prefix (as defined for your user ID) is used as the high-level qualifier for any name that is not enclosed in quotes.

The auxiliary file must be a REUSEable RRDS file, whose record length definitions are compatible with the file being edited.

If this field is left blank, a RRDS file is defined for the edit session and subsequently deleted.

Parent panels

- "Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- None.

Set Trace options panel

The Set Trace options panel allows you to view and modify parameters that control the Trace output produced by File Manager when it is run in debug mode.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                  Set Trace options

Trace Options:
   Trace destination
   3  1. FMNTRC
      2. Terminal
      3. Data set

   Data set name  JOHNLEV.FMN.TRACE
   Output class . A
   Trace limit .  500









Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Reset     F7=Backward
 F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 237. Set Trace options panel

Trace destination
Specifies where debug trace File Manager output is to be sent:

FMNTRC
Send trace output to the current FMNTRC allocation.

Terminal
Send trace output to the terminal.

Data set
Send trace output to the temporary data set specified by the **Data set name** field.

Data set name
The temporary trace data set where File Manager trace output is directed when the Data set trace destination is specified.

Output class
The class of the JES spool queue to be used when the PRINT command is issued when browsing the temporary trace data set.

Trace limit
When run in debug mode, File Manager writes the Trace output to a buffer (as for normal debug mode), but the buffer is only written to the Trace destination, if File Manager abnormally terminates.

This field specifies the number of Trace statements that are retained in the buffer. If you specify a value of zero, background debug mode is turned off.

Parent panels

"Set Processing Options panel" on page 661

Child panels

None.

Related tasks and examples

Tape Browse panel

The Tape Browse panel is used to specify the tape or tapes to be browsed, the number of files to be viewed and the starting position and display format to be used when browsing tape contents.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                      Tape Browse

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Files . . . . . 1____         number of files or EOV
   Initial BSF . . YES           NO to start from current tape position
   ASCII data  . . NO            YES to translate ASCII data for display









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 238. Tape Browse panel

Tapes: Lists the tapes currently allocated.

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Data Copy Functions panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager              Tape Data Copy Functions

1  Tape to tape  Copy tape volume to tape volume
2  Tape reformat Copy and reformat single tape file
3  Multifile     Copy multivolume and multiple tape files
4  To VSAM       Copy from tape to VSAM data set
5  To QSAM       Copy from tape to QSAM data set
6  From VSAM     Copy from VSAM data set to tape
7  From QSAM     Copy from QSAM data set to tape
8  ESV           Copy Exported Stacked Volume









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 239. Tape Data Copy Functions panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

- "Tape to Tape panel" on page 681
- "Tape to Tape Reblocked panel" on page 683
- "Tape to Labeled Tape panel" on page 680
- "Tape to VSAM panel" on page 685
- "Tape to QSAM panel" on page 684
- "VSAM to Tape panel" on page 728
- "Sequential Data to Tape panel" on page 648
- "Exported Stacked Volume Copy panel" on page 552

Related tasks and examples

Tape Label Display panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                  Tape Label Display

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Label printout  LONG          specify SHORT to only print VOL1 and
                                 first HDR label set









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 240. Tape Label Display panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Map panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                          Tape Map

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Print Options:
   Files . . . . . 1____        number of tape files, EOV, or EOD
   Blocks  . . . . 1_____       number of blocks
   Print format  . HEX__        CHAR or HEX









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 241. Tape Map panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Positioning Functions panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager              Tape Positioning Functions

1  BSF           Backspace file
2  FSF           Forward space file
3  BSR           Backspace record
4  FSR           Forward space record
5  REW           Rewind
6  RUN           Rewind and unload









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 242. Tape Positioning Functions panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Print panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                        Tape Print

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Print Options:
   Print format  . CHAR         CHAR or HEX
   Record format . U            F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . 50           if deblocking desired, fixed format only
   Records . . . . ALL          limit number of records to be printed
   Files . . . . . 1            number of tape files, EOV, or EOD




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 243. Tape Print panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Record Load panel

Panel and field definitions

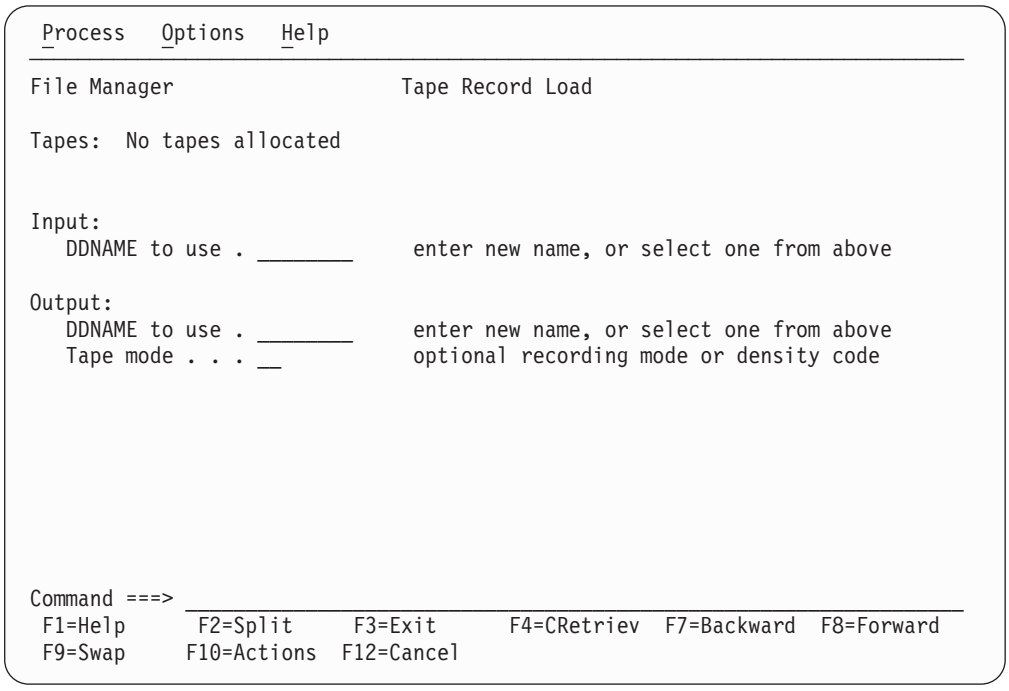


Figure 244. Tape Record Load panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Record Scan panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                    Tape Record Scan

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Record format . U___          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . 50_____       if deblocking desired, fixed format only

Scan Options:
   Pattern scan  . NO__          YES or NO
   Scan position . 1____         byte position within record to start at
   Scan argument . _____________________________________________________
   Number of hits  1_____        or ALL to list all scan hits in the file
   Records . . . . ALL___        limit number of records to be scanned



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 245. Tape Record Scan panel

Parent panels

"Tape Specific Functions panel"

Child panels

Related tasks and examples

Tape Specific Functions panel

The Tape Specific Functions panel is a menu panel that allows you to select the various tape-specific functions.

Panel and field definitions

```
Process   Options   Help
_____________________________________________________________________________

File Manager              Tape Specific Functions

1  Browse        Browse tape data
2  Copy          Copy tape data
3  Update        Copy and update tape data
4  Alter         Copy and alter tape data
5  Print         Print tape data
6  Contents      Summarize tape contents
7  Create        Create tape data
8  Summary       Print label summary
9  Compare       Compare two tapes
10 Scan          Scan tape data
11 Tape marks    Write tape marks
12 Initialize    Initialize a tape
13 Erase         Erase tape data
14 ESV list      List Exported Stacked Volume
15 Control       Position a tape



Command ===> _____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
F9=Swap     F10=Actions  F12=Cancel
```

Figure 246. Tape Specific Functions panel

1 Browse
Displays the Tape Browse panel, in which you can specify the tape records to be displayed for browsing.

2 Copy
Displays the Tape Data Copy Functions panel, which is another menu panel listing the various copy functions available for tape data.

3 Update
Displays the Tape Update panel, in which you can specify the tape records to be updated.

4 Alter
Displays the Tape Record Load panel, in which you can specify the tape records to be altered.

5 Print
Displays the Tape Print panel, in which you can specify the tape records to be printed.

6 Contents
Displays the Tape Map panel, in which you can specify the tape records to be included in a summary.

7 Create
Displays the Create Tape Data panel, in which you can specify details of the records to be created on a tape.

8 Summary
Displays the Tape Label Display panel, in which you can specify the tape labels and tape label summary to be printed.

9 Compare
Displays the Tape to Tape Compare panel, in which you can specify the tapes to be compared.

10 Scan
Displays the Tape Record Scan panel, in which you can specify the search conditions to be used when scanning the nominated tape.

11 Tape marks
Displays the Write Tape Mark panel, in which you can specify the number of Tape Marks to be written at the current position in the tape.

12 Initialize
Displays the Initialize Tape panel, in which you can start the process of initializing a labeled or unlabeled tape.

13 Erase
Displays the Erase Tape panel, in which you can specify the tape from which data is to be erased.

14 ESV list
Displays the Exported Stacked Volume List panel, in which you can specify an Exported Stacked Volume and a number of listing options for which a Table of Contents is to be printed.

15 Control
Displays the Tape Positioning Functions panel, which is another menu panel listing the various tape positioning functions available for tape data.

Parent panels

"Primary Option Menu panel" on page 615

Child panels

- "Tape Browse panel" on page 670
- "Tape Data Copy Functions panel" on page 671
- "Tape Update panel" on page 686
- "Tape Record Load panel" on page 676
- "Tape Print panel" on page 675
- "Tape Map panel" on page 673
- "Create Tape Data panel" on page 513
- "Tape Label Display panel" on page 672
- "Tape to Tape Compare panel" on page 682
- "Tape Record Scan panel" on page 677
- "Write Tape Mark panel" on page 737
- "Initialize Tape panel" on page 586
- "Erase Tape panel" on page 551
- "Exported Stacked Volume List panel" on page 553
- "Tape Positioning Functions panel" on page 674

Related tasks and examples

Tape to Labeled Tape panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                Tape to Labeled Tape

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above
   Type of tape  . SL___        SL or MIXED (mixture of SL and NL files)

Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Files . . . . . ALL__        number of files to be copied or ALL
   New Volser(s) . ____________________________  or * or blank




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 247. Tape to Labeled Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape to Tape panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                      Tape to Tape

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above

Output:
   DDNAME to use . ________      enter new name, or select one from above
   Tape mode . . . __            optional recording mode or density code
   Files . . . . . 1_____        number of tape files, EOV, EOT, or EOD









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 248. Tape to Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape to Tape Compare panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Tape to Tape Compare

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

   DDNAME to use . _______      enter new name, or select one from above

   Files . . . . . 1____        number of tape files, EOV, or EOD









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 249. Tape to Tape Compare panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape to Tape Reblocked panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                Tape to Tape Reblocked

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above
   Record format . U___         F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . 50_____      if deblocking desired, fixed format only

Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Record format . ___          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Block size  . . _______      required for blocked output




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 250. Tape to Tape Reblocked panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape to QSAM panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager              Tape to Sequential Data

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Record format . U___          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . 50_____       if deblocking desired, fixed format only
   Skip count  . . ________      number of records to be skipped
   Copy count  . . ALL_____      number of records to be copied
   Files . . . . . 1____         number of tape files, EOV, or EOD

Output:
   Data set name . ____________________________________________
   Member  . . . . ________      if partitioned
   Volume serial . ______        if not cataloged
   Unit  . . . . . ________      for tape data sets
   Disposition . . CAT           OLD, MOD, NEW, or CAT (for NEW,CATLG)
   Record format . FB__          if new format: U,F,V, or D, with B,S,A,M
   Block size  . . _____         physical output block size
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 251. Tape to QSAM panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape to VSAM panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                      Tape to VSAM

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________      enter new name, or select one from above
   Record format . U___          F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Record length . 50_____       if deblocking desired, fixed format only
   Skip count  . . ________      number of records to be skipped
   Copy count  . . ALL_____      number of records to be copied
   Files . . . . . 1_____        number of tape files, EOV, or EOD

Output:
   Data set name . ____________________________________________
   Reuse . . . . . NO___         YES or NO



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 252. Tape to VSAM panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Tape Update panel

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                      Tape Update

Tapes:  No tapes allocated


Input:
   DDNAME to use . ________     enter new name, or select one from above

Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Files . . . . . 1____        number of files or EOV









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 253. Tape Update panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Template and Copybook Utility functions panel

Panel and field definitions

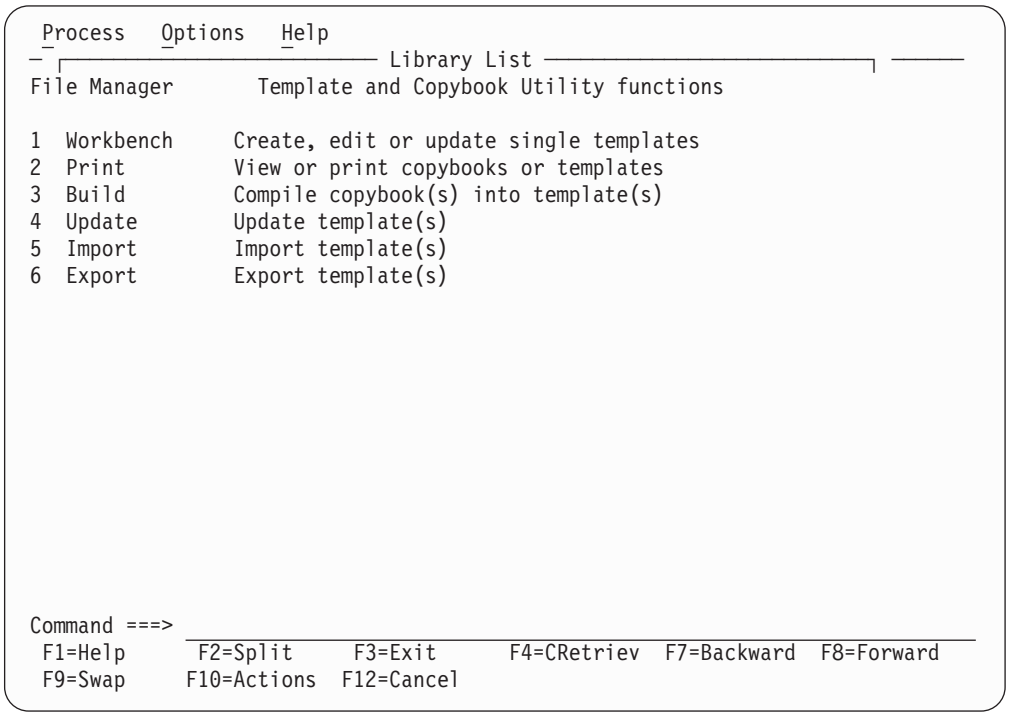


Figure 254. Template and Copybook Utility functions panel

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

- "Template Workbench panel" on page 698
- "Copybook View and Print: Entry panel" on page 507
- "Template Build Utility panel" on page 688
- "Template Update Utility panel" on page 693

Related tasks and examples

Template Build Utility panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager           Template Build Utility

Copybook:
   Data set name .  ___________________________________________
   Member  . . . .  _________  Blank or pattern for member list

Template:
   Data set name .  ___________________________________________
   Member mask . .  ________

Processing Options:
   Enter "/" to select option
   _  Batch execution
   _  Replace existing templates
   _  Advanced member selection
   _  Skip member list
   _


Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 255. Template Build Utility panel

Data set name (Copybook)
Must be a PDS(E) except for vendor-managed library systems such as Panvalet, CA/Librarian, and so on.

Member (Copybook)
You can enter a member name or pattern or leave blank.

Data set name (Template)
Must be a PDS or PDSE identifying where the templates will be stored.

Member mask
Name of a member of a partitioned data set. If you leave the member name blank, or specify a member name pattern, File Manager displays a member name list. You can then select the required member by entering S in the Select field for the appropriate member.

Batch execution
Enter "/" to edit the JCL to run the function in batch.

Replace existing templates
Enter "/" to replace like named members in the output partitioned data set.

Advanced member selection
Enter "/" to specify a range of members to be selected, rather than a specific or generic member name. If you specify this option, the copybook member name is ignored.

Skip member name list
Enter "/" to run without showing the member selection list. This option is ignored if errors are found whilst processing (for example, duplicate output member names when an output member mask has been used).

Parent panels
- "Template and Copybook Utility functions panel" on page 687

Child panels
- "Member Selection panel" on page 600

Related tasks and examples
- "Creating corresponding templates from selected copybooks" on page 155

Template Export Utility panel

The Template Export Utility panel provides a means of exporting templates to external XML templates.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager           Template Export Utility
 Command ===> ____________________________________________________________

 Template:
    Data set name . 'FMN.IMP.EXAMPLE'_____________________________
    Member  . . . . ________ (Blank or pattern for member list)
    Filter  . . . . __________ __________ __________ __________

 Export Data set:
    Data set name . 'FMN.XML.EXAMPLE'_____________________________
    Member mask . . ________

 Processing Options:
    Enter "/" to select option            Enter "/" to select option
    _  Batch execution                       Replace members
    _  Advanced member selection          /  Copybook and criteria only
    _  Skip member list                   _  Stats On
    _
```

Figure 256. Template Export Utility entry panel

(Template) Data set name (required)
A fully qualified or generic data set name that is used to identify the data set where the template(s) to be exported reside

Member
You can enter a member name, or pattern, or leave blank. If you select advanced member selection, then this name constitutes the first name on the member range panel.

Filter Enter up to 4 member names or patterns that are used as a filter for the template selection list that is displayed. The filter name is created as a copybook name for base and IMS templates. For IMS views and criteria it is treated as an originating template. Any other type of template is skipped if a filter is specified.

(Export Data set) Data set name (Required)
A fully qualified or generic data set name that is used to identify the XML template data set where the exported XML templates are stored.

Member mask
This is a rename mask and allows the exported templates to be stored under an alternative name.

Specify a pattern to rename members in the output partitioned data set based on the member names in the input partitioned data set. A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent sign (%).

* A place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter `ABC*` the renamed members all begin with "ABC" followed by the remainder of the old member name.

% A place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter `%%%A*M` the first three characters of the renamed members remain unchanged, the fourth character is replaced with the letter "A", and the remainder of the old member name are unchanged.

Batch execution
Allows you to edit batch JCL to run the function in batch.

Advanced member selection
Allows you to specify a range of members to be selected, rather than a specific or generic member name.

Skip member list
Runs without showing the member selection list. This option bypasses the member list panel and processes all the qualifying members.

Replace members
Replaces like-named members in an output partitioned data set.

Copybook and criteria only
Restrict the output XML to only the copybook and criteria tags.

Stats On
Always update or create ISPF statistics for exported members.

```
 Process   Options   Help
 ------------------------------------------------------------------------------
  File Manager            Template Member Selection          Row 00001 of 00002
 Command ===> _____________________________________________________ Scroll CSR

 Input data set  FMN.IMP.EXAMPLE
 Export data set FMN.XML.EXAMPLE

        Name     Prompt   Type Created    Updated             Lang   Ver Descr _
        *        *        *    *          *                   *      *   *
 ______ CONVT1            DB2  2012/09/19 2012/09/19 11:40:56 NONE     3
 ______ DJ1E              IMS  2012/09/19 2012/09/19 10:44:11 COBOL    3
 ****  End of data  ****
```

Figure 257. Template Member Selection export member list

Sel Prefix area. You can enter the following prefix commands:

B Performs an ISPF BROWSE command against the member. The member is displayed using the ISPF Browse panel, not the File Manager Browse panel.
E Invoke template editing.
S Select for template update.
U Invoke the template source definition edit and update process for the template.

Name Name of the member.

Prompt
Status values relevant to the action being performed.

Created
The date the template was created.

Updated
For templates, this is the last changed date. For IMS views and extract criteria, this is the last time this was updated from its originating template

Lang The original language of the copybooks from which the fields in the template were derived.

Ver The version or level of the template.

Description
(To view, scroll right with the Right function key, F11.) The first line of the description that can be entered using the DESCRIBE command during template editing.

```
 Process   Options   Help
 File Manager            Template Member Selection          Row 00001 of 00002
Command ===>                                                  Scroll CSR

Input data set  FMN.IMP.EXAMPLE
Export data set FMN.XML.EXAMPLE

-      Name     SSID Owner    Object name        DBD data set                  _
       *        *    *        *                  *
       CONVT1   DFB2 DSN8810  EMP
       DJ1E                                     FMIMS.XTEST.DBDLIB(DJ1E)
****  End of data  ****
```

Figure 258. Scrolling right on the member list

SSID Subsystem ID for DB2 template
Owner DB2
Object owner for DB2 template
Object name
DB2 object name for DB2 template
DBD data set
DBD data set name for IMS template, view or criteria set

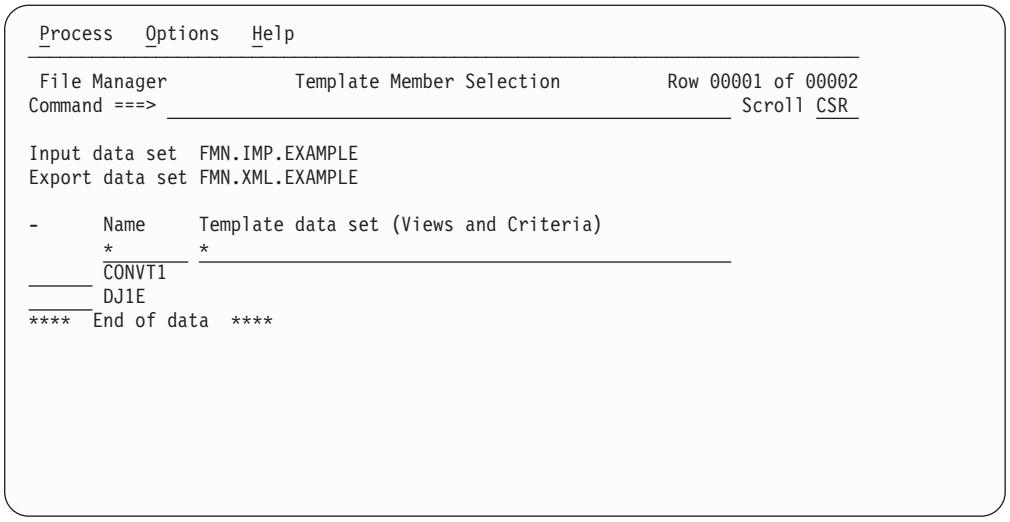


Figure 259. Scrolling further right on the member list

Template data set
Originating template data set for IMS view or criteria set.

Template Import Utility panel

The Template Import Utility panel provides a means of importing external XML templates.

Panel and field definitions

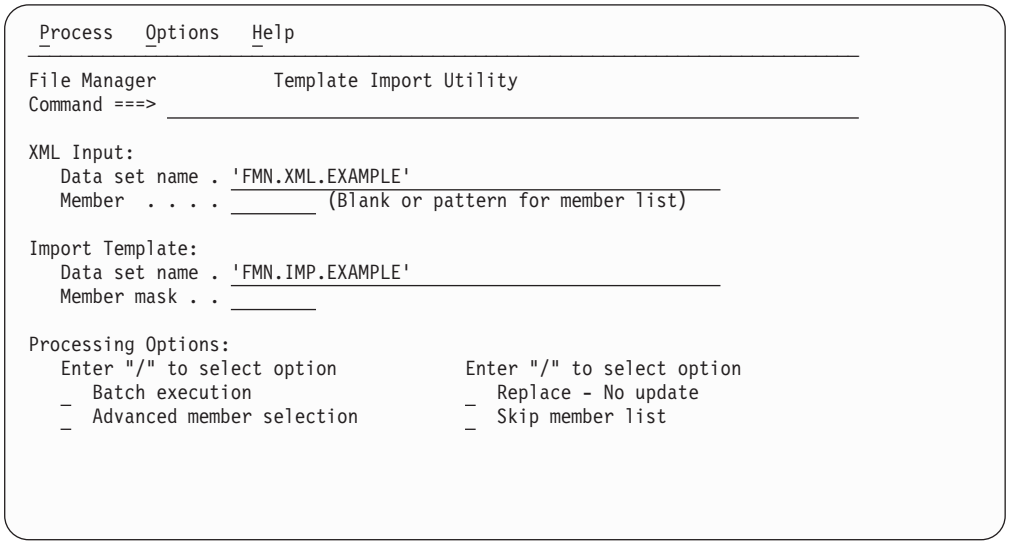


Figure 260. Template Import Utility entry panel

(XML Input) Data set name (Required)
A fully qualified or generic data set name that is used to identify the XML input data set.

Member
You can enter a member name, or pattern, or leave blank. If you select advanced member selection, then this name constitutes the first name on the member range panel.

(Import Template) Data set name (Required)
A fully qualified or generic data set name that is used to identify the template data set where the imported templates is stored.

Member mask
This is a rename mask and allows the exported templates to be stored under an alternative name.

Specify a pattern to rename members in the output partitioned data set based on the member names in the input partitioned data set. A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent sign (%).

* A place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter `ABC*` the renamed members all begin with "ABC" followed by the remainder of the old member name.

% A place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter `%%%A*` the first three characters of the renamed members remain unchanged, the fourth character is replaced with the letter "A", and the remainder of the old member name are unchanged.

Batch execution
Allows you to edit batch JCL to run the function in batch.

Advanced member selection
Allows you to specify a range of members to be selected, rather than a specific or generic member name.

Replace - No update
Replaces like-named members in an output partitioned data set. If this option is not selected and the member exists, it is updated with the XML template information.

Skip member list
Runs without showing the member selection list. This option bypasses the member list panel and processes all the qualifying members.

Template Update Utility panel

There are two forms of the Template Update Utility panel, the entry panel (shown in Figure 261 on page 694) and the member list panel (shown in Figure 262 on page 696).

The member list panel is displayed after you have specified the required details on the entry panel and pressed Enter.

Panel and field definitions

```
 Process   Options   Help
─ ┌──────────────────────── Library List ──────────────────────────┐ ──────
File Manager          Template Update Utility                  Top of data

Template:
   Data set name . ___________________________________________
   Member  . . . . ________     Blank or pattern for member list
   Copybook filter __________  __________  __________  __________
Output Template:
   Data set name . ___________________________________________
   Member mask . . ________
Processing Options:
   Enter "/" to select option          Enter "/" to select option
   _  Batch execution                  _  Replace members
   _  Advanced member selection        _  Use library data sets
   _  Skip member list                 _  Check mode - no update
   _  Preserve copybook library
Library data sets:
Data set name 1 ___________________________________________
              2 ___________________________________________
              3 ___________________________________________
              4 ___________________________________________
              5 ___________________________________________
              6 ___________________________________________
              7 ___________________________________________
              8 ___________________________________________
              9 ___________________________________________
             10 ___________________________________________
             11 ___________________________________________
             12 ___________________________________________
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 261. Template Update Utility - entry panel

Template

Data set name
Data set name. Must be a PDS(E).

Member
You can enter a member name, or pattern, or leave blank. If you select advanced member selection, then this name constitutes the first name on the member range panel.

Copybook filter
Up to 4 member names or patterns used as a filter so that only templates referencing those copybooks, or copybooks that match the patterns, are selected for processing.

Output Template
This is an optional data set and can be used to provide an alternative data set for the resultant updated templates.

Data set name
Must be a PDS or PDSE identifying where the update templates are to be stored.

Member mask
This is a rename mask and allows the updated templates to be stored under an alternative name.

Specify a pattern to rename members in the output partitioned data set based on the member names in the input partitioned data set. A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent sign (%).

*
A place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs.

For example, if you enter:

```
ABC*
```

the renamed members will all begin with ABC followed by the remainder of the old member name.

%
A place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name.

For example, if you enter:

```
%%%A*
```

the first 3 characters of the renamed members remain unchanged, the fourth character is replaced with the letter "A", and the remainder of the old member name are unchanged.

Processing Options

Batch execution
Allows you to edit batch JCL to run the function in batch.

Replace members
Replaces like-named members in an output partitioned data set.

Advanced member selection
Allows you to specify a range of members to be selected, rather than a specific or generic member name.

Use library data sets
Uses the library data sets specified on the entry panel.

Skip member list
Runs without showing the member selection list. This option bypasses the member list panel and processes all the qualifying members. If an error (like a duplicate output name) results from the specified parameters, then the member list panel is displayed with the errors highlighted.

Preserve copybook library
Ensures that, if a copybook still exists in the library that it was previously found in and that library is in the list that the update is using, then that version of the copybook is used.

If you do not select this option, or the copybook no longer exists in the library it was previously found in, or that library is not in the list the update is using, then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

Check mode - no update
Runs the function without saving the resulting members. Normally used to check for errors.

Library data sets
These fields are only used when the **Use library data sets** option is selected. If selected, the copybook library names referenced in the template are changed to copybook library names specified in these fields. The new library data sets are used to locate and compile the copybooks during the update process, so you need to ensure that all the copybooks referenced in all the templates selected are available in the libraries you specify to avoid update errors occurring.

```
 Process   Options   Help
 ┌──────────────────────── Library List ──────────────────────────┐ ──────
File Manager              Template Update Utility               Row 1 of 104

Input data set  'FMN.RFM0047.TEMPLATE'
Update data set 'FMN.RFM0047.TEMPLATE'

Sel Name     Prompt       Created     Updated              Lang   Ver
_   ABEND                 2004/02/11  2004/02/11 09:09:15  COBOL    1
_   BIGCHAR               2004/07/14  2006/06/28 16:12:31  COBOL    1
_   BIGCOPY               2003/12/17  2003/12/17 14:39:16  COBOL    1
_   BIGCOP2               2004/02/11  2004/02/11 10:12:10  COBOL    1
_   BIGKSDS               2001/08/10  2001/08/16 13:54:23  COBOL    1
_   BUGGY                 2004/06/24  2004/06/25 10:56:12  COBOL    1
_   CJM01A                2004/06/15  2004/06/15 12:59:06  COBOL    1
_   CNOTFND               2006/07/18  2006/07/18 13:59:25  COBOL    1
_   COMPERR               2006/06/09  2006/06/22 09:45:08  COBOL    1
_   COPYTST               2006/01/10  2006/07/07 08:28:28  COBOL    1
_   COPY01                2006/02/28  2006/07/26 23:31:29  COBOL    1
_   COPY0102              2005/05/03  2006/07/26 23:31:32  COBOL    1
_   CRITERR               2006/07/18  2006/07/18 15:08:00  COBOL    1
Command ===> ____________________________________________________ Scroll CSR
 F1=Help    F2=Split   F3=Exit    F4=Expand  F5=RFind   F7=Up      F8=Down
 F9=Swap   F10=Left   F11=Right  F12=Cancel
```

Figure 262. Template Update Utility - member list

Sel Prefix area. You can enter the following prefix commands:
- **B** Performs an ISPF BROWSE command against the member. The member is displayed using the ISPF Browse panel, not the File Manager Browse panel.
- **E** Invoke base template editing.
- **S** Select for template update.
- **U** Invoke the template source definition edit and update process for the template.

Name Name of the member.

Prompt
Status values relevant to the action being performed.

Created
The date the template was created.

Updated
For templates, this is the last changed date. For IMS views and extract criteria, this is the last time this was updated from its originating template

Lang The original language of the copybooks from which the fields in the template were derived.

Ver The version or level of the template.

Description
(To view, scroll right with the Right function key, F11.) The first line of the description that can be entered using the DESCRIBE command during template editing.

Parent panels

- "Template and Copybook Utility functions panel" on page 687

Child panels

- None.

Related tasks and examples

- "Updating one or more templates" on page 160

Template Save pop-up panel

The Template Save panel allows you to save a temporary template. It is only displayed when you have created a new copybook or dynamic template and then exit from the template editing panels.

Panel and field definitions

```
                          Template Save
_
F
C   To discard changes to your template, press PF12/Cancel.

    To use changes without saving, press PF6/RunTemp.

    To save changes, type a data set and member name below, then press
    PF3/Exit.

    Data set name  _____________________________________________
    Member  . . .  ________




    Command ===> ___________________________________________________________
C    F1=Help       F2=Split      F3=Exit       F6=RunTemp    F7=Backward
     F8=Forward    F9=Swap      F12=Cancel
```

Figure 263. Template Save pop-up panel

Data set name
The name of an existing data set, into which you want to save your template. Can be a fully-qualified data set name or a pattern. The name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

Member

The name of a new member to be created in the existing data set, which will contain your new template definitions. If you specify an existing member name, the existing member is replaced by your new template. If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Parent panels

- "Record Type Selection panel" on page 636 (when saving a copybook template with two or more record types)
- Figure 174 on page 566 (when saving a copybook template with one record type)
- "Dynamic Template panel" on page 529 (when saving a dynamic template)

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Specifying or updating the source definition for a copybook template" on page 146
- "Creating dynamic templates" on page 152

Template Workbench panel

The Template Workbench is a central area from which you can create, edit and maintain your templates. It can be accessed from the File Manager Primary Options menu (Option **7. Templates**), or by issuing the **TView** primary command on either the View or the Edit panel.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Template Workbench

CC Create template from copybook       E  Edit field/record in template
CM Create template from model          U  Update template from copybook
MC Map from copybook                   MT Map from template

Copybook:
   Data set name . COPYBK3_____________________________________________
   Member  . . . . ________
Template:
   Data set name . 'USERID.TEMPLATE'___________________________________
   Member  . . . . COPY01__
Model Template:
   Data set name . TEMPLATE____________________________________________
   Member  . . . . TUTDYN__

Processing Options:
 Enter "/" to select option
 /  Advanced copybook selection









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Describe  F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F12=Cancel
```

Figure 264. The Template Workbench panel from the Primary Options Menu

Note: If the Template Workbench is accessed from an editor session, two additional options (**RC - Run using copybook** and **RT Run using template**) appear on the panel.

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                 Template Workbench

RC Run using copybook                  RT Run using template
CC Create template from copybook       E  Edit field/record in template
CM Create template from model          U  Update template from copybook
MC Map from copybook                   MT Map from template

Copybook:
   Data set name . 'FMNUSER.COPYBOOK'_______________________________
   Member  . . . . COPY01____
Template:
   Data set name . 'FMNUSER.TEMPLATE'_______________________________
   Member  . . . . TEMP0102
Model Template:
   Data set name . _____________________________________________________
   Member  . . . . ________

Processing Options:
 Enter "/" to select option
 /  Advanced copybook selection
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RT        F6=Describe
 F7=Backward  F8=Forward   F9=Swap      F10=Actions  F12=Cancel
```

Figure 265. The Template Workbench panel from an editor session

Template Workbench commands

The workbench commands that you can use are listed at the top of the panel. You can position the cursor on the required command and press Enter or you can type them on the Command Line and press Enter. These are:

RC The RC command is used to run the current function using a temporary template created from a copybook. The temporary template cannot be edited before the function is run, and is not saved. Use the CC command if you want to edit or save the template.

You must specify the PDS member name of the copybook you want to use, and the data set name or PDS member name of the template you want to update. The copybook you specify is validated by compiling it. For a description of this process, and information about what you can do if errors occur, see the description of the CC command. This command is only available when the workbench has been invoked from an editor session using the TEdit or TView commands.

RT The RT command is used to run the current function with an existing template. The template can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function using the CC command. This command is only available when the workbench has been invoked from an editor session using the TEdit or TView commands.

CC The CC command is used to create a template from a copybook. You can edit the template before running a function with it. You can also save the template for later use.

You must specify the PDS member name of the copybook you want to use. The copybook you specify is validated by compiling it. File Manager includes the copybook in a shell program, and invokes the compiler to syntax-check the copybook, and to produce an ADATA file. If the compilation completes without errors, the information in the ADATA file is processed to create a template. If the compilation completes with errors, a pop-up menu is displayed. From the pop-up menu you can choose to:

- View the compilation listing.
- Abort the template creation process.
- Retry the compilation. Before you select this option, you should view the compilation listing and correct any errors in the copybook. While you are viewing the compilation listing, you can use File Manager's split screen facility to swap to another session and use your editor to correct the errors in the copybook.

If File Manager is able to create a template and you provide the name of a sequential data set or PDS member in which the template is to be saved, File Manager saves it for you to use later with other functions. If you do not provide a template name, the template is kept in storage until the end of the current function. You can still use other Template Workbench commands to work with the template, but it is not automatically saved. However, if, at any time before the end of the current function, you decide to save the template, you can specify a template name and save it using the SAVE command.

CM The CM command is used to create a new template by copying an existing template. You must provide the data set name or PDS member name of the existing template. If you provide the name of a data set or PDS member in which the new template is to be saved, File Manager saves it for you to use later with other functions. If you do not provide a template name, the template is kept in storage until the end of the current function. You can still use other Template Workbench commands to work with the template, but it is not saved for later use. If, at any time before the end of the current function, you decide to save the template, you can specify a template name and save it using the SAVE command.

E The E command is used to edit an existing template. The template can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function. You can create a temporary template by using the CC command. Alternatively, you can let File Manager automatically create one for you. To do this, specify the member name of a COBOL copybook, but do not specify a template data set name or member name. Then, when you issue the E command, File Manager automatically creates a temporary template.

When you edit a template, a series of panels is displayed on which you can specify the following information:

- For a file containing multiple record types, which record types you want to be selected for processing.
- For a file containing multiple record types, what criteria File Manager should use to identify each record type.
- If you only want to process certain records, such as those containing specified values, what criteria File Manager should use to select those records.
- If you don't want to process all fields within a record, which fields you do want to process.
- What headings you want to use for fields when they are displayed on the screen or printed using the SNGL or TABL display or print formats.
- For numeric fields, whether you want leading zero suppression when they are displayed on the screen or printed using the SNGL or TABL display or print formats.
- When you use the template to help generate test data, what data patterns File Manager should use for each field.

MC The MC command is used to map the records and fields in one COBOL copybook to the records and fields in another copybook or in a template. You can use the MC command when you want to copy all records in one data set to another data set, and you want to reorganize the fields in the “From” data set before they are copied to the “To” data set. Do not use the MC command if you want to select which records in the “From” data set you want to copy. Instead, you should first create a template, edit it to specify the record selection criteria, and then use the MT command to do the field mapping.

You must specify the PDS member name of the COBOL copybook you want to use. This copybook describes the contents of the “From” data set. The copybook you specify is validated by

compiling it with the COBOL compiler. For a description of this process, and information about what you can do if errors occur, see the description of the CC command.

If File Manager is able to create a template from the copybook, another panel is displayed where you specify the name of a COBOL copybook or the name of a template that describes the contents of the "To" data set. From this panel you can specify how you want the fields in the "From" data set mapped to the "To" data set.

MT The MT command is used to map the records and fields in one template to the records and fields in another template or in a COBOL copybook. You can use the MT command when you want to copy all or selected records in one data set to another data set, and you want to reorganize the fields in the "From" data set before they are copied to the "To" data set.

The template can be either a permanent one previously saved in a data set or PDS member, or a temporary one created for the current function using the CC command. This template describes the contents of the "From" data set, and the record selection criteria for the records you want to copy. If you want to use a previously saved template, you must specify the data set name (and member name, if required) of the template.

When you use the MT command, another panel is displayed where you specify the name of a template or the name of a COBOL copybook that describes the contents of the "To" data set. From this panel, you can specify how you want the fields in the "From" data set mapped to the "To" data set.

U The U command is used to update an existing template from a copybook. You should use this function when you have made minor changes to the copybook, and you want these changes reflected in the template you previously created from the copybook. If you have made significant changes to the copybook, the update process might not produce the results you expect, so you should instead use the CC command to recreate the template. Minor changes that the update process supports include:

- Changing field names without changing field data types
- Changing field data types without changing field names
- Changing the order of fields in a record
- Deleting unreferenced fields
- Inserting new fields
- Changing record length
- Changing the number of occurrences of fields in an array (table)

Major changes that might cause the update process to produce unwanted results include:

- Changing field names and field data types
- Changing field names and the order of fields in a record

Copybook
The name (data set and member) of the copybook that contains the record description of your data.

Template
The name (data set and, if required, member) of an existing template you want to run with or edit, or a new template you want to create.

Model Template
The name (data set and, if required, member) of an existing template you want to use as a model for creating a new template.

Advanced copybook selection
When selected, specifies that you want to:
- Use one or more copybooks to create your template
- Be able to specify how these copybooks are interpreted by File Manager

This option applies to the CC and U commands and is provided to enable a template to be generated from more than one copybook which may be located in one or more libraries. If this option is selected, a number of panels may be displayed.

Parent panels

- "Template and Copybook Utility functions panel" on page 687
- "Browse panel" on page 455 (by issuing the TVIEW command)
- "Editor panel" on page 534 (by issuing the TVIEW command)

Child panels

- "Library List panel" on page 588
- "Map To panel" on page 595
- "Record Type Selection panel" on page 636
- "Field Selection/Edit panel" on page 565

Related tasks and examples

Utility Functions menu panel

Lists the utility functions available within File Manager. Selecting an option displays the relevant utility Entry panel.

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                  Utility Functions

0  DBCS          Set DBCS data format for print
1  Create        Create data
2  Print         Print data
3  Copy          Copy data
4  Dslist        Catalog services
5  VTOC          Work with VTOC
6  Find/Change   Search for and change data
7  AFP           Browse AFP data
8  Storage       Browse user storage
9  Printdsn      Browse File Manager print data set
10 Loadlib       View load module information
11 Compare       Compare data
12 Audit trail   Print audit trail report
13 Copybook      View and Print
14 Websphere MQ  List Websphere MQ managers and queues




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 266. Utility Functions menu panel

0 DBCS
Displays the Set DBCS Format panel

1 Create
Displays the Data Create Utility panel

2 Print
Displays the Print Utility panel

3 Copy
Displays the Copy Utility panel

4 Dslist
Displays the Catalog Services panel

5 VTOC
Displays the Display VTOC panel

6 Find/Change
Displays the Find/Change Utility panel

7 AFP Displays the AFP Print Browse panel

8 Storage
Displays the Memory Browse panel

9 Printdsn
Displays print output in full-screen mode

10 Loadlib
Displays the Load Module Information panel

11 Compare
Displays the Compare Utility panel

12 Audit trail
Displays the Print Audit Trail panel

13 Copybook
Displays the Copybook View and Print panel

14 WebSphere MQ
Displays the Websphere MQ Managers panel

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

- Set DBCS Format panel
- Data Create Utility panel
- Print Utility panel
- Copy Utility panel
- Catalog Services panel
- Display VTOC panel
- Find/Change Utility panel
- AFP Print Browse panel
- Memory Browse panel
- Print Browse panel
- Compare Utility panel
- Print Audit Trail panel
- Copybook View and Print panel
- MQ Manager Selection List panel

Equivalent functions

- None.

Related tasks and examples

See the "Related tasks" section for each child panel.

Value List Edit panel

This panel allows you to to create and edit value lists to be used in field scrambling.

Panel and field definitions

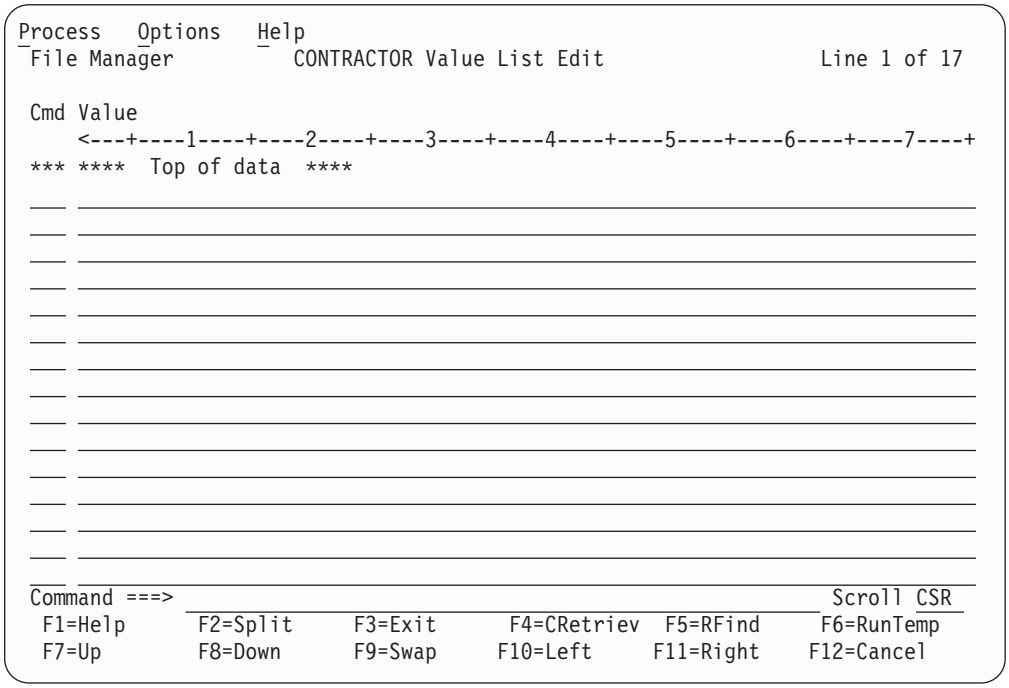


Figure 267. Value List Edit panel

Cmd Prefix command field. Allows you to copy, insert, move, repeat or delete lines. Available commands are:

- **A** Identifies the line after which data is to be moved or copied.
- **B** Identifies the line before which data is to be moved or copied.
- **C** Copy one line.
- **C***n* Copy *n* lines.
- **CC** Copy block of lines. Marks the start and end of the block.
- **D** Delete one line.
- **D***n* Delete *n* lines.
- **DD** Delete block of lines. Marks the start and end of the block.
- **I** Insert one empty line.
- **I***n* Insert *n* empty lines.
- **M** Move one line.
- **M***n* Move *n* lines.
- **MM** Move block of lines. Marks the start and end of the block.
- **R** Repeat line once.
- **R***n* Repeat line n times.
- **RR** Repeat block of lines. Marks the start and end of the block.
- **RR***n* Repeat block of lines *n* times. Marks the start and end of the block.

Value A value that can be used when field scrambling is in effect.

Parent panels

- "Field Attributes panel - alphanumeric fields" on page 553
- "Field Attributes panel - numeric field" on page 559

Related tasks and examples

- "Specifying and editing a value list" on page 184
- "Scrambling data" on page 262

View panel

You use the View panel to display a selected data set or data set member, scroll through the records and find and, if necessary, temporarily change specific information within the records. You can also use this panel to temporarily insert or delete new records, join or split existing records and copy or repeat records within the data set.

The View panel displays different fields, depending upon the type of data set shown and whether or not a template has been used.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
View             FMN.V11R1M0.SFMNSAM1(FMNCDATA)               Rec 33 of 40
       Col 1          Insert length 80                          Format CHAR
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
****** ****  Top of data  ****
000001        01 REC-TYPE01.
000002           03 REC-TYPE              PIC XX.
000003           03 NAME                  PIC X(20).
000004           03 EMPLOYEE-NO           PIC 9(4) BINARY.
000005           03 AGE                   PIC 9(4) BINARY.
000006           03 SALARY                PIC 9(7) PACKED-DECIMAL.
000007           03 MONTH
000008                        PIC 9(8) BINARY OCCURS 12 TIMES.
000009           03 FILLER                PIC XX.
****** ****  End of data  ****









Command ===> ____________________________________________________________ Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 268. View panel showing text file

Title The Title identifies the function (Edit) and the data set being used. For a PDS or PDSE member, the data set name includes the member name. Record number information, or short messages, appear to the right.

Col The Col field shows the column number of the column that is currently at the far left of the data area. You can scroll to a specific column (left or right) by typing a new value. The default value for this field is 1.

Insert length
The Insert length field specifies the initial length of inserted records. The possible range of values is from 1 to 9999999, depending on the data set

characteristics. The default value is the maximum record length as defined in the catalog. When a User I/O exit is being used, the insert length value is adjusted to the maximum length accepted by the exit for the data set being edited.

Scale The Scale shows the columns of the data area.

Prefix area (line numbers)
Area (if displayed) displaying the record number of each line in the data set. This area also doubles as the prefix command entry area.

You can use the following prefix commands when editing:

A Identifies the record after which data is to be moved or copied.

A "K" can be appended to the A to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

B Identifies the record before which data is to be moved or copied.

A "K" can be appended to the B to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

BND (Available only in CHAR, HEX and LHEX display formats.) Displays a bounds line on the following line, indicating the columns searched by the CHANGE, FIND, and EXCLUDE commands. To change the bounds, type a < on the bounds line to define the left bound and a > to define the right bound. To remove the bounds line from the display, use the D prefix command or the RESET SPECIAL primary command.

C Copy one record.

C*n* Copy *n* records.

CC Copy block of records. Mark start and end of block.

COLS Displays the column identification line (CHAR, HEX, and LHEX display formats only).

D Delete one record.

D*n* Delete *n* records.

DD Delete block of records. Mark start and end of block.

F Display the first record of a block of excluded records.

F*n* Display the first *n* records of a block of excluded records.

I Insert one empty record.

If the display format is CHAR, HEX or LHEX, then the entire record is initialized to blanks.

If the display format is TABL, then the record is initialized according to the record structure of the current record type:

- Numeric fields are initialized to zero.
- Alphanumeric fields are initialized to blanks.

- If the record type contains a variable-length array, then the record is initialized with the minimum number of array elements, and the array size field (or fields, for multi-dimensional arrays) is initialized accordingly.

I*n* Insert *n* empty records.

L Display the last record of a block of excluded records.

L*n* Display the last *n* records of a block of excluded records.

LC Translate all uppercase characters in a record contents to lower case (see **Caution** in LCC).

LC*n* Translate all uppercase characters in *n* records to lower case (see **Caution** in LCC).

LCC Translate all uppercase characters in a block of records to lower case. Mark start and end of block.

Caution: The LC commands (LC, LC*n*, LCC) affect all characters in a record, not just characters in those fields with an alphanumeric or character data type. This means that numeric data, such as binary data or packed decimal data, can be corrupted by using these commands.

M Move one record.

M*n* Move *n* records.

MM Move block of records. Mark start and end of the block.

O Overlay one record.

O The O (overlay) command indicates the target when you want to merge lines. Overlay is used in conjunction with move and copy commands.

A "K" can be appended to the O to indicate that the source is to be used with multiple destinations. The last destination must not contain the K append so that File Manager knows the command set is complete. No part of the command set is executed until the command set is complete.

O*n* Overlay *n* records.

OO Overlay block of records. Mark start and end of the block.

P Identifies the record preceding which data is to be moved or copied. This is synonymous with the B prefix command.

R or " Repeat one record.

R*n* **or "***n*
Repeat *n* records.

RF Perform a refresh of the record. (Shared files only.)

RF*n* Perform a refresh of *n* records. (Shared files only.)

RFF Perform a refresh of a block of records. Mark start and end of the block. (Shared files only.)

RR or ""
Repeat block of records. Mark start and end of the block.

RR*n* **or ""***n*
Repeat block of records *n* times. Mark start and end of block.

SV Perform a save of the record. (Shared files only.)

SV*n* Perform a save of *n* records. (Shared files only.)

SVV Perform a save of a block of records. Mark start and end of the block. (Shared files only.)

UC Translate all lowercase characters in a record to upper case (see **Caution** in UCC).

UC*n* Translate all lowercase characters in *n* records to upper case (see **Caution** in UCC).

UCC Translate all lower case characters in a block of records to upper case. Mark start and end of block.

Caution: The UC commands (UC, UC*n*, UCC) affect all characters in a record, not just characters in those fields with an alphanumeric or character data type. This means that numeric data, such as binary data or packed decimal data, can be corrupted by using these commands.

V (Available only when you are using a template.) Display currently suppressed records of the record type indicated by the shadow line. Records of other record types are suppressed from display.

X Exclude record from display. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position. To show an excluded record, use the F or LA prefix command.

X*n* Exclude *n* records from display. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position.

XX Exclude block of records from display. Mark start and end of block. If the display of excluded record shadow lines is enabled (see "SHADOW primary command" on page 816), a shadow line is shown indicating how many records are being excluded at this position.

.*xxxx* Label a line. The label, *xxxx*, is a string of 1 to 4 alphabetic characters, that does not start with the character "Z" (labels starting with "Z" are system labels). Labels can be assigned to any data line. Labels cannot be assigned to the bounds line or to shadow lines.

) Shift record right one position.[1]

)*n* Shift record right *n* positions.[1]

)) Shift block of records right one position. Mark start and end of block.[1]

))*n* Shift block of records right *n* positions. Mark start and end of the block.[1]

(Shift record left one position.[1]

(*n* Shift record left *n* positions.[1]

((Shift block of records left one position. Mark start and end of block.[1]

((*n* Shift block of records left *n* positions. Mark start and end of block.[1]

/ Position record at the top of the screen.

1. The shift occurs only for that part of the record within the current bounds limit. See "Setting bounds" on page 109 for a detailed explanation.

Data Area
The Data Area shows the data in the selected display format. For a description of the different display formats, see "Selecting a display format" on page 66.

Command
The Command line is a field in which you can enter Primary Commands, such as CHANGE.

Scroll The Scroll field defines the current scroll amount. You can type a new value.

When a VSAM data set is displayed, additional fields can be seen.

```
Edit            FMN.REQ77.RRDS                                  Rec 0
       Type RRDS                                               Format CHAR
       Col 1          Insert length 4089
       ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
****** ****  Top of data  ****
000001 xxxxxxyyy orig 1
```

Figure 269. Edit panel showing VSAM data set

Type The type of VSAM data set, for example, RRDS. IAM files are also indicated.

When a data set is displayed in TABL format with a template, the column headings show the field names defined in the template.

```
Edit            FMN.V11R1M0.SFMNSAM1(FMNCDATA)              Rec 13 of 40
                                                            Format TABL
       REC-TYPE NAME                 EMPLOYEE-NO     AGE   SALARY  MONTH(1)
       #2       #3                         #4       #5       #6        #7
       AN 1:2   AN 3:20                BI 23:2 BI 25:2  PD 27:4   BI 31:4
       <>       <---+----1----+---->     <---+>  <---+> <---+--> <---+--->
****** ****  Top of data  ****
000001 01       GRAHAM JONES              5512      94    68000         7
```

Figure 270. PDSE data set member in TABL format

When a data set is displayed in SNGL format with a template, the field and record number of the current record is shown.

```
Edit            FMN.V11R1M0.SFMNSAM1(FMNCDATA)               Rec 13 of 40
Current type is REC-TYPE01                                       Format SNGL
                                  Top Line is 1    of 32    in Record 3
Ref Field    Picture Typ Start   Len   Data
  2 REC-TYPE
              XX      AN      1     2   01
  3 NAME      X(20)   AN      3    20   Graham Jones
  4 EMPLOYEE-NO
              9(4)    BI     23     2    5512
  5 AGE       9(4)    BI     25     2    94
  6 SALARY    9(7)    PD     27     4    68000
```

Figure 271. PDSE data set in SNGL format

Available commands

- "RESET primary command" on page 806
- "RFIND primary command" on page 809
- "RIGHT primary command" on page 810
- "SAVE primary command" on page 812
- "SAVEAS primary command (templates)" on page 813
- "SHADOW primary command" on page 816
- "SHOW primary command" on page 817
- "SLOC primary command" on page 818
- "SORT primary command" on page 818
- "SPLT primary command" on page 821
- "SPLTJOIN primary command" on page 821
- "STR primary command" on page 821
- "TEDIT primary command" on page 824
- "TOP primary command" on page 824
- "TVIEW primary command" on page 825
- "TYPE primary command" on page 826
- "UP primary command" on page 826
- "VIEW primary command" on page 828
- "ZOOM primary command" on page 829

Parent panels

- "View Entry panel"

Child panels

- "Template Workbench panel" on page 698 (to invoke the panel, enter the TVIEW primary command).
- "Record Type Selection panel" on page 636 (to invoke the panel, enter the TEDIT primary command - panel displays when template contains more than one record type).
- "Field Selection/Edit panel" on page 565 (to invoke the panel, enter the TEDIT primary command - panel displays when using a copybook template with only one record type).
- "Dynamic Template panel" on page 529 (to invoke the panel, enter the TEDIT primary command - panel displays when using a dynamic template).
- "Record Selection Criteria panel" on page 631 (to invoke the panel, enter the CEDIT primary command).
- "Record Identification Criteria panel" on page 625 (to invoke the panel, enter the CEDIT ID primary command).

Equivalent functions

- "DSV (Data Set View)" on page 998

Related tasks and examples

- Chapter 3, "Viewing and changing data sets," on page 49

View Entry panel

You use the View Entry panel to select a data set for viewing in the View panel.

Panel and field definitions

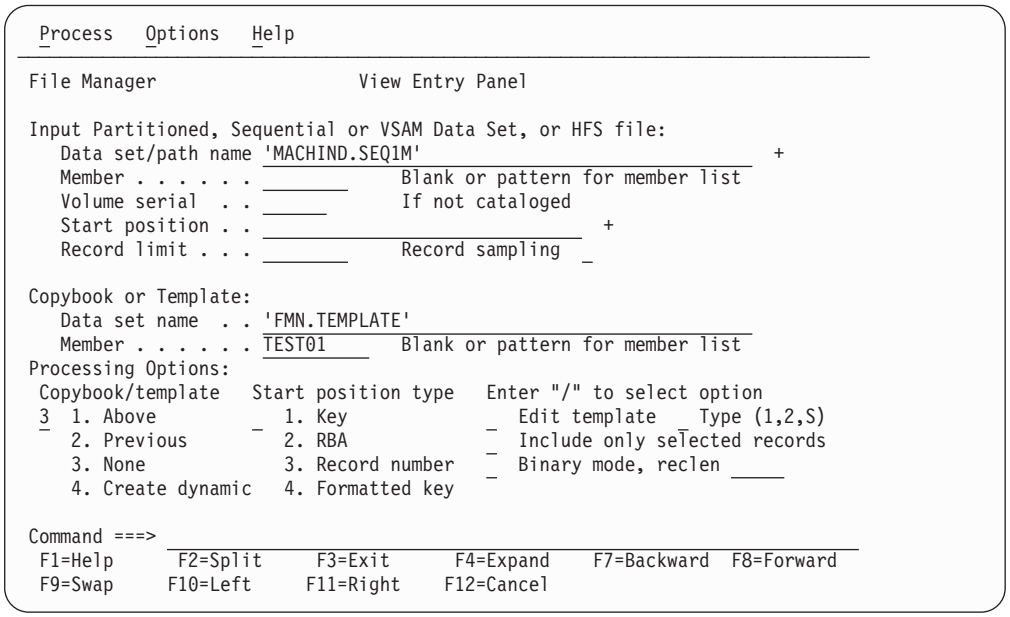


Figure 272. View Entry panel

Data set/path name
Can be a fully-qualified data set name or a pattern, an HFS file or directory, a WebSphere MQ queue name, or a CICS resource.

For information about specifying a WebSphere MQ queue name, see "Specifying an MQ manager or queue" on page 21.

For information about specifying a CICS resource, see "Specifying a CICS resource" on page 21.

The data set name may include a member name or name pattern in parenthesis. If the member is specified here, the associated **Member** field must be empty.

When you specify an HFS file or directory, you must enter a full path name. If the path name is longer than the displayed entry field, press the Expand function key (F4) to display a pop-up window in which you can enter a longer name.

Member
If you specified the name of a partitioned data set (PDS) without including a member name or name pattern in parenthesis in the **Data set name** field, then you can use this field to specify the member name or a member name pattern.

Volume serial
Serial number of the volume which contains the data set. Required for data sets which are not cataloged.

Start position
Initial starting position for the data set to be viewed. The initial display is positioned at the specified record.

The default is the top of the data set. You can enter a negative record to indicate the number of records before the end of file. For example, to see just the last record on the file, enter -1 as the start point.

The format of the start position field is either numeric or character, depending upon the type of start position selected. For VSAM KSDS Key values, if the value you enter contains leading, embedded or trailing blanks, commas or quotation marks, it must be enclosed in quotation marks. You may also enter a hexadecimal string enclosed in quotation marks and preceded by X or x, for example, `X'C1C2'`. The maximum number of characters, including any required characters, is 250.

When your cursor is in this field, you can use the LEFT/RIGHT commands to scroll within the field (function keys F10/F11). You can also use the EXPAND function (F4), to open the key field in a window.

Note: The Erase EOF key only works on the displayed part of the key. When the key being displayed is larger than the field area on the screen, you must either scroll or expand the field to erase the unseen portions.

For all other data set formats, a valid unsigned number must be entered.

To specify a starting position as the number of records before the end of file, enter a negative record number. For example, to see just the last record on the file, enter a start position of -1.

If you specify a starting position, you should also specify the **Start position type** in the Processing Options. If you do not, File Manager assumes that the value provided in the Starting position field is a record number.

VSAM - KSDS: Key value
If the key is not matched, the record with a key greater than the value given is specified. If the key value is greater than the last record in the data set, the initial position is the "End of data" marker.

VSAM - ESDS: RBA (Relative Byte Address) Value
If the RBA is not matched, the record with a RBA greater than the value given is specified. If the RBA value is greater than that of the last record in the data set, the initial position is the "End of data" marker.

VSAM - RRDS: Slot value
If the slot number is greater than the last used slot in the data set, the initial position is the "End of data" marker.

QSAM: Record Number
If the Record Number is greater than that of the last record in the data set, the initial position is the "End of data" marker.

HFS As for QSAM.

Record limit
This field restricts the number of records retrieved from a data set (from the start point or top) resulting in an edit or view of a portion of the data set. You can use the keyword "MEMORY" as a record limit to restrict the number of records retrieved to as many as will fit comfortably in the available virtual storage.

The default start position is the top of the data set.

Record Sampling
Indicates whether you want record sampling to be performed on the data set. If you select this option, File Manager displays the Record Sampling panel.

Copybook or Template
Data set name and Member name of the template or copybook to be used to format your data.

Note: In the case of a copybook, this can be the name of a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Copybook/template usage
Indicates if you want to use a template for a logical view of the data.

1 Use the template specified on the panel (or compile the specified copybook into a template, and use that).

2 Use the last template associated with the data set.

3 No logical view is to be used by the function.

4 Create a dynamic template.

Start position type
Determines how File Manager interprets the value provided in the Starting position field. If the Start position type is not specified, File Manager assumes that the value is a Record Number.

1. KEY
Only valid when the data set is a VSAM KSDS, VSAM AIX or VSAM PATH.

2. RBA
Only valid when the data set is a VSAM KSDS, VSAM AIX or VSAM ESDS.

3. Record Number
Default. Valid for any type of supported data set.

4. Formatted key
Only valid when the dataset is a VSAM KSDS, VSAM AIX, or VSAM PATH and when a template is provided. This displays the formatted key positioning panel.

Note: For templates with more than one layout, a 01 selection list is displayed first.

Edit template
Indicates if you want to edit the template before use. You edit the template when you need to change format, selection, data create attributes, and reformatting information for output copy templates.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Include only selected records
When the editor session is either in-storage (a record limit of “MEMORY” is used, or record limit is omitted and a record limit of "MEMORY" is

implied by means of this option) or when you have selected **Record Sampling**, only the records selected by template processing are included in the editor session.

This option does not alter the operation of editing or viewing if a template is not used.

Binary mode
When processing an HFS file, allows you to specify binary mode (selected) or text mode (unselected).

reclen When processing an HFS file and the **Binary mode** option is selected, records are derived based on the fixed record length specified. The default is 80.

Can be in the range: 1–32760

Use I/O exit
Allows you to specify a user I/O exit for compressed or encrypted data sets.

This option has two fields. To select the option, enter "/" in the field to the left of Use I/O exit. With this selected, you can then specify which exit to use in the field to the right of the field label.

Note:

1. The fields only display if File Manager is installed with the option **USEIOX=ENABLE**, and the **Exit enabled** field (in the Set System Processing Options panel) is set to YES. If a default is specified with either of those options, it is displayed in the field to the right of Use I/O exit.
2. I/O exits can only be used to process the data sets that you are using. They cannot be used to process the copybook or template that you are using to format the data set.

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

Depending upon the options selected in the View Entry panel, the next panel displayed can be:

- "Editor panel" on page 534
- "Data Set Selection panel" on page 519 (a pattern has been entered in the Data set name field)
- "Member Selection panel" on page 600(a pattern or a blank has been entered in the Member field)
- "Record Type Selection panel" on page 636 (Edit Template is selected and the specified template was based on a copybook with more than one record type).
- "Field Selection/Edit panel" on page 565 (Edit Template is selected and the specified template was based on a copybook with only one record type).
- "Dynamic Template panel" on page 529 (Create Dynamic option is selected or Edit Template is selected and the specified template was created dynamically).
- "Record Sampling panel" on page 628 (Record Sampling option selected)

- "Personal Data Set List panel" on page 612 (Current Data Set List option selected from the Process drop-down menu, or REFL fastpath command entered).
- "Personal Data Set Lists panel" on page 614 (Personal Data Set Lists option selected from the Process drop-down menu, or REFD fastpath command entered).

Equivalent functions

- "DSV (Data Set View)" on page 998

Related tasks and examples

- "Supplying a procedure when using a File Manager panel" on page 417
- "Starting an editor session without using templates" on page 51

Volume Summary/Selection panel

Lists volumes which are currently online. To find a specific volume in the list, you can scroll forward and backward through the list, or you can use the LOCATE command. To select a volume, or multiple volumes, enter S in the Select field(s) for the volume(s), and press Enter. You can also select all volumes by entering ALL on the command line and pressing Enter. You can deselect any selected volume by overtyping the S in its Select field.

Panel and field definitions

```
 Process   Options   Help
 File Manager            Volume Summary/Selection           Row 00001 of 00005
Unit    *        DSN       'TYRONED.FMDATA.**'
DevType *        VOLSTATE  ALL         SMS SG *
Volumes 5        Data sets 7           VSAM   3           non-VSAM 4
VOLSER  D$US5*   Trks used N/A         Free   N/A         Utilized N/A
  VOLSER UNIT Dtype SMS-SG   Total  Used   %Used Free trk Tot DSN VSAM   nVSA ±
  D$US5* *    *     *
_ D$US51 E801 3390  PRIMARY  150255 128750    86    21505       1           1
_ D$US53 E81C 3390  PRIMARY  150255 142702    95     7553       1           1
_ D$US55 E929 3390  PRIMARY  150255 141320    94     8935       1           1
_ D$US57 EA52 3390  PRIMARY  150255 127706    85    22549       2      2
_ D$US58 EA55 3390  PRIMARY  150255 131406    87    18849       2      1    1
****  End of data  ****

Command ===>                                                    Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F6=Process
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 273. Volume Summary/Selection panel

Sel Selection field in which you can enter one of the following line commands:

S Selects a volume. Displays the Display VTOC Data Set List panel.

I Displays detailed volume information on the Disk Volume Details panel.

Parent panels

- "Display VTOC panel" on page 523

Related tasks and examples

- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Manipulating your view of selection lists" on page 28

VSAM Association Information panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager          VSAM Association Information

VSAM Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.KSDS8'
   Catalog ID . . . . 'CATALOG.SYSPLEXD.USER'

VSAM Associations:
   Path . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.PATH1'
   Path . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.PATH2'

   AIX  . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1'
     Data component . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1.DATA'
     Index component  'SYATES.UNITTEST.RFM0189.KSDS8.AIX1.INDEX'
     Path . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1PTH1'
     Path . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX1PTH2'

   AIX  . . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX2'
     Data component . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX2.DATA'
     Index component  'SYATES.UNITTEST.RFM0189.KSDS8.AIX2.INDEX'
     Path . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX2PTH1'
     Path . . . . . . 'SYATES.UNITTEST.RFM0189.KSDS8.AIX2PTH2'
                      ****  End of data  ****
Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Info      F7=Up
 F8=Down      F9=Swap     F10=Actions   F12=Cancel
```

Figure 274. VSAM Association Information panel

Parent panels

Child panels

Related tasks and examples

VSAM Define panel

You can use the VSAM Define panel to specify the allocation attributes for a VSAM data set. This panel has five variations, one for each VSAM data type supported by File Manager (KSDS, ESDS, RRDS, VRRDS, LDS). For general information about defining data sets, see the *DFSMS Using Data Sets* manual appropriate for your operating system. For specific information about valid field values, use File Manager's Field-level Help (put your cursor in the field and press F1).

Panel and field definitions

```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                      VSAM Define

VSAM Catalog Entry:
   Data set name  . . 'USERID.NEWDS'
   Catalog ID . . . .

Basic Information:
   VSAM data type . . KSDS        Expiration date  . (NONE)
   Allocate EAS . . .             Blank, NO or OPT
   Data . . . . . .
   Index  . . . . . .
VSAM Cluster Attributes:
   Key length . . . .             Key offset . . . .
   CI size  . . . . .             size of the data control intervals
   Buffer space . . .             buffer space to be allocated at open time
   Shr cross region .             cross system . . .             Reuse . . . .
   Recovery . . . . .             Spanned  . . . . .             Erase . . . .
   Writecheck . . . .             BWO  . . . . . . . NO          LOG . . .  NONE
   Logstreamid  . . . USERID.ABC
VSAM Data Allocation:
   Allocation unit  . CYL         REC, KB, MB, TRK, or CYL
   Space primary  . . 1           secondary  . . . . 0
   Recsize average  . 4089        maximum  . . . . . 4089
   Freespace  % CI  . 0           % CA . . . . . . . 0
   Volume serial(s) . MV8W31
VSAM Index Allocation:
   CI size  . . . . . 1024        size of the index control intervals
   Allocation unit  . TRK         REC, KB, MB, TRK, or CYL
   Space primary  . . 1           Secondary . . . 0
   Volume serial(s) . MV8W31
SMS Definitions:
   Data class . . . . *UNKNOWN
   Storage class  . . BASE
   Management class . STANDARD
Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Up        F8=Down
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 275. VSAM Define panel for KSDS

Parent panels

"Allocate panel" on page 448

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

"Allocating a new data set" on page 246

VSAM Edit Sharing Options panel

The VSAM Edit Sharing Options panel presents options which apply only when editing VSAM files that are potentially being shared by other users.

This is the case for the File Manager base product when a KSDS SHAREOPTIONS 3 or SHAREOPTIONS 4 file is edited. It is also the case when a (non-KSDS) SHAREOPTIONS 3 or SHAREOPTIONS 4 file can not be allocated to the TSO session as DISP=OLD. Instead, the file is allocated DISP=SHR to allow access to the file and the edit continues with the restriction of inplace editing mode. Alternatively, this could be a CICS file accessed through the File Manager CICS product. In either of the above cases, these sharing options become active. These options are designed to provide improved useability for those files that are in such a shared environment.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager              VSAM Edit Sharing Options


These options apply only when editing VSAM files that are being shared.


 Autosave frequency              ________

Options for the CHANGE ALL command
 SAVE before starting            ___  Enter ON or OFF.
 Change all autosave frequency   ________
 Auto retry for change all       ___  Enter ON or OFF.









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 276. VSAM Edit Sharing Options panel

Autosave frequency
For shared files, issue a AUTOSAVE command after a given number of changes. Set this value to 0 if you do not want an AUTOSAVE command to be issued.

SAVE before starting
When operating on a shared file, during a CHANGE command with the ALL operand, this option determines if a SAVE is issued before commencing the change.

Setting this ON ensures that any updates made to the file so far are saved. During the CHANGE ALL process, when using AUTORTRY (and a CHGASAVE value), the change process may need to refresh records as it goes (due to concurrent updates by other users). That refresh may discard other updates that had been made in this edit session to records on the file.

To issue a SAVE before the CHANGE commences, and therefore ensure the state of the records that have been updated, leave the setting at ON.

Change all autosave frequency
This option applies specifically to the operation of the CHANGE command when used with the ALL parameter. Similar to the AUTOSAVE option above, it provides the ability to save records after a given number of changes. If used, the changes are counted at the record level. That is, if a record is changed (regardless of how many times the CHANGE command changed that record), it is counted as one change. When the number of records changed reaches the number set in this option, a SAVE command is issued and the CHANGE command continues.

Auto retry for change all
This option applies specifically to the operation of the CHANGE command when used with the ALL parameter. This option controls the action taken when a SAVE command is issued during operation of the CHANGE command and another user has updated a record while the change was running. If this option is set ON, the CHANGE command refreshes (that is, it rereads) the record that this occurred on and attempts to reapply the change.

If this option is OFF, each time File Manager detects that the record was updated before the CHANGE command saved the record, File Manager displays a popup window that gives you the choice of updating the record anyway, refreshing the record and reapplying the update, or aborting the CHANGE command.

Parent panels

"Set Processing Options panel" on page 661

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Working with File Manager and shared files" on page 89
- "VSAM edit sharing options (option 0.7)" on page 45

VSAM Entry Detail panel

The Catalog Services: VSAM Entry Detail panel is used to display the parameters of a selected VSAM data set, allowing you to review or modify them. The information displayed varies according to the type of entry.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                VSAM Entry Detail                    Top of data

VSAM Catalog Entry:
   Data set name  . . 'SYATES.UNITTEST.RFM0189.CLUSTER.KSDS1'
   Catalog ID . . . . 'CATALOG.USER3.SYSPLEXD'

Additional information available with Stats(F11) and ASsocs(F6) commands.

Basic Information:
   VSAM data type . . KSDS         KSDS, ESDS, RRDS, VRRDS or LDS
   Creation date  . . 2003/12/17 (351)
   Expiration date  . ***None***
   Last modified date ***None***

   Data component . . 'SYATES.UNITTEST.RFM0189.DATA.KSDS1.DATA'
   Index component  . 'SYATES.UNITTEST.RFM0189.INDEX.KSDS1.INDEX'

VSAM Associations:
   Path . . . . . .
   AIX  . . . . . .
     Data component .
     Index component
     Path . . . . . .

SMS Definitions:                                Backup/Logging:
   SMS managed  . . . Y                         BWO . . .
   Data class . . . . *UNKNOWN                  Log . . .
   Storage class  . . PRIMARY                   Logstreamid
   Management class . PRIMARY
   Last backup date . 0000.000.0000

VSAM Cluster Attributes:
   CI size  . . . . . 20480    size of the data control intervals
   Buffer space . . . 41984    buffer space to be allocated at open time

   Share options .    Cross region . 1          Cross systems  3
   Process options    Reuse  . . . . N          Recovery . . . Y
                      Spanned  . . . N          Erase  . . . . N
                      Extended . . . N          Extended addr  N
                      Compressed . . N          Writecheck . . N

VSAM Data Allocation:
   Allocation unit  . CYL      REC, KB, MB, TRK, or CYL

   Space . . . . .    Primary . . 1             Secondary . 0
   Record size . .    Average . . 4089          Maximum . . 4089
   Free space . .     % of CI . . 0             % of CA . . 0
   Number of volumes  . . . . . . 1      Tracks allocated . 15

   Volume serial(s) . D$US01
   Device type(s) . . 3390-3

VSAM Key Definition:
   Key length . . . . 64               Key offset  . . 0

VSAM Index Allocation:
   CI size  . . . . . 1024     size of the index control intervals
   Allocation unit  . TRK      REC, KB, MB, TRK, or CYL

   Space . . . . .    Primary . . 1             Secondary . 0
   Number of volumes  . . . . . . 1      Tracks allocated . 1

   Volume serial(s) . D$US01
   Device type(s) . . 3390-3
Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=Volumes   F6=Assocs
 F7=Up        F8=Down      F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 277. VSAM Entry Detail panel (Example showing KSDS data set)

Parameters for a VSAM entry

These panels show catalog information such as:

- The VSAM data set type
- The creation date
- The expiration date
- The date and time last modified
- VSAM associations (cluster, AIX, and path)
- The SMS attributes
- Backup/Logging options
- The control interval size
- The buffer space to be allocated at open time
- The share options
- Reuse (Y or N)
- Recovery (Y or N)
- Spanned (Y or N)
- Erase (Y or N)
- Extended (Y or N)
- Extended addressed (Y or N)
- Compressed (Y or N)
- The space unit for allocation
- The number of units for primary and secondary allocation
- The average and maximum record size
- Free space, as a percentage of the control interval and control area
- Total volumes and tracks allocated for data and index components
- Up to five volume serials and device types
- The key length
- The position of the key within the record (starting with 1)
- Whether the key must be unique (Y or N)
- Up to five key ranges (specified as hexadecimal numbers)

Note: For most of these parameters, you can use the Help function key (F1) to obtain an explanation about the parameter.

Parameters for a path

If you select a path, you can see:

- The creation date
- The expiration date
- The associated alternate index
- The associated cluster

Parameters for a non-VSAM entry

If you select a non-VSAM entry, you can see:

- The creation date
- The expiration date listed in the catalog
- Up to five volume serials and device types
- The SMS attributes

Note: If you change the expiration date for a non-VSAM entry with Catalog Services, this change does not affect the expiration date in the disk VTOC or tape header. Only the catalog entry is changed.

If the data set is a GDG (generation data group) member or has aliases, you can also see:

- The GDG base
- The alias or aliases

Parameters for a GDG base

If you select a GDG base, you can see:
- The creation date
- The expiration date
- The maximum number of GDG members
- The names of the first five GDG members

Parameters for an OAM object collection

If you select an OAM object collection, you can see:
- The creation date
- The expiration date
- The SMS attributes
- The directory token

Parameters for an alias

If you select an alias, you can see the user catalog or data set to which the alias points.

Parameters for a non-VSAM entry

If you select a non-VSAM entry, you can see:
- The creation date
- The expiration date listed in the catalog
- Up to five volume serials and device types
- The SMS attributes

Note: If you change the expiration date for a non-VSAM entry with Catalog Services, this change does not affect the expiration date in the disk VTOC or tape header. Only the catalog entry is changed.

If the data set is a GDG (generation data group) member or has aliases, you can also see:
- The GDG base
- The alias or aliases

Parameters for a GDG base

If you select a GDG base, you can see:
- The creation date
- The expiration date
- The maximum number of GDG members
- The names of the first five GDG members

Parameters for an OAM object collection

If you select an OAM object collection, you can see:
- The creation date
- The expiration date
- The SMS attributes
- The directory token

Parameters for an alias

If you select an alias, you can see the user catalog or data set to which the alias points.

Parent panels

- "Catalog Services Data Set List panel" on page 466

Child panels

- None.

Equivalent functions

- None.

Related tasks and examples

- "Managing catalog entries" on page 311

VSAM Entry Rename panel

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager                  VSAM Entry Rename

VSAM Catalog Entry:
   Data set name  . . 'TYRONED.FMDATA.KSDS'
   Catalog ID . . . . 'CATALOG.USER3.SYSPLEXD'

Additional information available with Stats(F11) and ASsocs(F6) commands.

Basic Information:
   VSAM data type . . KSDS         KSDS, ESDS, RRDS, VRRDS or LDS
   Data component . . 'TYRONED.FMDATA.KSDS.DATA'
   Index component  . 'TYRONED.FMDATA.KSDS.INDEX'

To Rename a VSAM entity, type the new Data set name above and press ENTER.
The component names of a VSAM CLUSTER will be adjusted automatically based
on the newly entered data set name as outlined in the Help (F1).
If finer degree of control over the component names is required then the ALTER
command may be used to adjust component names manually.

Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=Volumes   F6=Assocs
 F7=Up        F8=Down      F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 278. VSAM Entry Rename panel

Parent panels

"Catalog Services Data Set List panel" on page 466 "Display VTOC Data Set List panel" on page 528

Child panels

Related tasks and examples

"Managing catalog entries" on page 311

VSAM Statistics and Extent Detail panel

The VSAM Statistics and Extent Detail panel displays the extents of a disk data set.

To display the VSAM Statistics and Extent Detail panel, press the Stats function key (F11) on the VSAM Entry Detail panel.

To locate specific data, use the LOCATE primary command.

Panel and field definitions

```
  Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────────
File Manager          VSAM Statistics and Extent Detail

VSAM Catalog Entry:
   Data set name . 'GREGCZ.VSTRIP2B.RRDS01'
   Catalog ID  . . 'CATALOG.USER1.SYSPLEXD'
                                                                        More:
VSAM Statistics:
 Component ---------------- Records ------------------ ------ Splits -------
  -------  ---Total-- -Deleted-- -Inserted- -Updated-- ----CI---- ----CA----
  Data           3300          0       3300          0          0          0

Data Extents:
Total volumes: 25 (15 used, 10 candidates)  Extents: 15  Tracks: 1650
 Ext Volume   ---Begin-end--- Reltrk,------  ----------- Kilobytes ----------
 num serial   Cyl-hd   Cyl-hd       numtrks   Low-alloc High-alloc  High-used
     D$ST13(1)  Extents: 1  Tracks: 110                     59400      59400
   1 --"--     68  0    75  4   1020,110              0      59399
     D$ST0C(2)  Extents: 1  Tracks: 110                     59400          0
   2 --"--     68  0    75  4   1020,110              0      59399
     D$ST07(3)  Extents: 1  Tracks: 110                     59400          0
   3 --"--    801 11   809  0  12026,110              0      59399
     D$ST14(4)  Extents: 1  Tracks: 110                     59400          0
   4 --"--     68  0    75  4   1020,110              0      59399
     D$ST02(5)  Extents: 1  Tracks: 110                     59400          0
   5 --"--    618 10   625 14   9280,110              0      59399
     D$ST06(6)  Extents: 1  Tracks: 110                     59400          0
   6 --"--    761 14   769  3  11429,110              0      59399
     D$ST11(7)  Extents: 1  Tracks: 110                     59400          0
   7 --"--     68  2    75  6   1022,110              0      59399
     D$ST18(8)  Extents: 1  Tracks: 110                     59400          0
   8 --"--     68  1    75  5   1021,110              0      59399
     D$ST08(9)  Extents: 1  Tracks: 110                     59400          0
   9 --"--    713  2   720  6  10697,110              0      59399
     D$ST0B(10)  Extents: 1  Tracks: 110                    59400          0
  10 --"--     68  1    75  5   1021,110              0      59399
     D$ST04(11)  Extents: 1  Tracks: 110                    59400          0
  11 --"--    763  0   770  4  11445,110              0      59399
     D$ST0A(12)  Extents: 1  Tracks: 110                    59400          0
  12 --"--    763  2   770  6  11447,110              0      59399
     D$ST09(13)  Extents: 1  Tracks: 110                    59400          0
  13 --"--    763  1   770  5  11446,110              0      59399
     D$ST03(14)  Extents: 1  Tracks: 110                    59400          0
  14 --"--    761 14   769  3  11429,110              0      59399
     D$ST01(15)  Extents: 1  Tracks: 110                    59400          0
  15 --"--    619 12   627  1   9297,110              0      59399
     *(16)      (candidate)
     *(17)      (candidate)
     *(18)      (candidate)
     *(19)      (candidate)
     *(20)      (candidate)
     *(21)      (candidate)
     *(22)      (candidate)
     *(23)      (candidate)
     *(24)      (candidate)
     *(25)      (candidate)

Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F6=Process   F7=Up
 F8=Down      F9=Swap     F10=Actions  F11=Info     F12=Cancel
```

Figure 279.

Parent panels

- "VSAM Entry Detail panel" on page 722

Child panels

Related tasks and examples

VSAM to Tape panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                      VSAM to Tape

Tapes:  No tapes allocated


Input:
   Data set name . ____________________________________________
   Start key . . . _____________________________ key or slot
   Skip count  . . ________      number of records to be skipped
   Copy count  . . ALL_____      number of records to be copied

Output:
   DDNAME to use . ________      enter new name, or select one from above
   Tape mode . . . __            optional recording mode or density code
   Record format . ___           F,FB, V,VB,VS,VBS, D,DB,DS,DBS, or U
   Block size  . . _______       required for blocked output



Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 280. VSAM to Tape panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

WebSphere MQ Functions panel

You use this panel to select a WebSphere MQ function.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager                Websphere MQ Functions

 1  List          List Managers and queues.
 2  View          View a WMQ queue.
 3  Edit          Edit a WMQ queue.









 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 281.

The functions you can select are:

1 List Displays the Websphere MQ Managers panel.

2 View
Displays the Websphere MQ Queue Editor Entry panel.

3 Edit As for **2 View**.

Parent panels

- "Primary Option Menu panel" on page 615

Child panels

- "WebSphere MQ Managers panel"
- "WebSphere MQ Queue Editor Entry panel" on page 732

Related tasks and examples

- "Working with WebSphere MQ" on page 331

WebSphere MQ Managers panel

This panel is used to select an item from a list.

You can use it, for example, to select an WebSphere MQ queue name that you want to view.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager               Websphere MQ Managers             Row 0001 of 0004

  SSID   ↓ Name     Active   Code     Reason
  <---+--> <---+--> <---+--> <---+--> <---+----10---+----2--->
_ CSQ1     CSQ1MSTR Active              2 MQRC_Q_MGR_NOT_AVAILABLE
_ CSQ2     CSQ2MSTR Active              2 MQRC_Q_MGR_NOT_AVAILABLE
_ CSQ3     CSQ3MSTR Active              2 MQRC_Q_MGR_NOT_AVAILABLE
_ CSQ4     CSQ4MSTR Active              2 MQRC_Q_MGR_NOT_AVAILABLE
_ MQ62     MQ62MSTR Active              2 MQRC_Q_MGR_NOT_AVAILABLE
_ MQ72     MQ72MSTR Active              0 MQRC_NONE
****  End of data  ****




Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 282.

Parent panels

- "WebSphere MQ Functions panel" on page 728

Child panels

- (with I prefix command)
- "WebSphere MQ Queue List panel" on page 736 (with S prefix command)

Related tasks and examples

- "Working with WebSphere MQ" on page 331

WebSphere MQ Managers Information panel

This panel provides the attributes of the queue manager that are typically displayed as a result of the DISPLAY QMGR ALL command, including a terse description of the attribute.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager           Websphere MQ Manager Information

Queue Manager:
   Manager name . . . . . . MQ72
   Description  . . . . . . IBM WebSphere MQ for z/OS - V7

Basic Information:
   Last altered date  . . . 2009-08-01     Time  . 06.33.19
   Platform . . . . . . . . MVS         CPI Level  100
   Command level  . . . . . 700

Queue and namelist names:
   Command queue  . . . . . SYSTEM.COMMAND.INPUT
   Dead letter queue  . . .
   Default XMIT queue . . .
   Repository name  . . . .
   Repository namelist  . .
   SSL Key Repository . . .
   SSL CRL Namelist . . . .

   SSL CRL Namelist . . . .

Security related:
   SSL Renegotiate  . . . . 0             SSL Server subtasks . . 0

Maximums:
   Priority . . . . . . . . 9             Message Length  . . . . 104857600
   Open Handles . . . . . . 256           Uncommitted messages  . 10000
   Channels . . . . . . . . 200           Out cluster channels  . 999999999
   Cluster exit data len  . 100           Active channels . . . . 200
   TCPIP Channels . . . . . 200           LU62 channels . . . . . 200

Events:
   Authorization events . . DISABLED      Local events  . . . . . DISABLED
   Remote events  . . . . . DISABLED      Command events  . . . . DISABLED
   Start/Stop events  . . . ENABLED       Config  events  . . . . DISABLED
   Inhibit events . . . . . DISABLED      Channel events  . . . . DISABLED
   Performance events . . . DISABLED      SSL events  . . . . . . DISABLED
   Bridge      events . . . DISABLED

Other:
   Syncpoint available  . . AVAILABLE     Expiry scan interval  . OFF
   Queue sharing group  . .               Trigger interval  . . . 999999999
   Channel autodef exit . .               Cluster workload exit .
   Cluster exit data  . . .
   Cluster put behaviour  . LOCAL         Shared queue mgr name . USE
   Intra group queueing . . DISABLED      Intra group authority . DEF
   Intra queue agent  . . .               IP Address version  . . IPV4
   Accounting collection  . ON            Q monitor collection  . OFF
   Channel collection . . . OFF           Auto cluster collect  . QMGR
   Trace route record . . . MSG           Activity record . . . .
   Adapter subtasks . . . . 9             Dispatchers . . . . . . 5
   TCPIP Keepalive used . . NO            TCPIP Task name . . . . TCPIP
   TCPIP Stack usage  . . . SINGLE        TCPIP Max port number . 0
   TCPIP Min port number  . 0             LU62 Generic name . . .
   LU62 outbound name . . .               LU62 APPCPM suffix  . .
   DDNS/WLM group . . . . .               Join WLM/DNS  . . . . . NO
   Listener restart time  . 60            Receive wait time . . . 0
   Receive wait type  . . . MULTIPLY      Receive min wait time . 0
   Adopt check elements . . ALL           Adopt restart orphans . ALL
   Channel trace size . . . 2             Channel trace start . . YES
Activity report generation  MSG           CCSID . . . . . . . . . 500
   Cluster date . . . . . .               . . . . . .
   IBM use only CHISERVP  . 00000000000000000000000000000000
Command ===>                                                 Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Each field on the panel has an associated field help panel containing text extracted from the Websphere MQ library, which remains the authoritative reference for all things MQ. Link here: http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp.

Parent panels

- "WebSphere MQ Managers panel" on page 729

Child panels

- None.

Related tasks and examples

- "Working with WebSphere MQ" on page 331

WebSphere MQ Queue Editor Entry panel

This panel is used to select an item from a list.

You can use it, for example, to select an WebSphere MQ queue name that you want to view.

Panel and field definitions

```
 Process   Options   Help
------------------------------------------------------------------------------
File Manager            Websphere MQ Queue Editor Entry

Websphere MQ Queue:
   Queue name . . . . D3.CSQ2.ANYQ                                  
   SSID . . . . . . . CSQ2
Include descriptors . _            (Message header information)
Copybook or Template:
   Data set name  . . ____________________________________________
   Member . . . . . . ________   (Blank or pattern for member list)
Processing Options:
 Copybook/template                        Enter "/" to select option
 3  1. Above                              _  Edit template    Type (1,2,S)
    2. Previous                           _  Include only selected records
    3. None                               _  Create audit trail
    4. Create dynamic




Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 284.

Queue name
The name of the WebSphere MQ queue you want to view.

For a generic queue name, use the percent % sign to represent a single character, and an asterisk * to represent any number of characters within the name.

When a name containing generic characters is used, a selection list of matching WebSphere MQ queues (on the associated WebSphere MQ queue manager) is presented.

SSID The name of the WebSphere MQ queue manager for the WebSphere MQ queue you want to view.

Include descriptors
When you select this option (with the / character), the message descriptor data returned by the GET MQ API is pre-pended to the record data, allowing these fields to be examined in conjunction with the message data.

If this option is not selected, only the message data is presented for viewing.

Data set name
A fully-qualified or generic data set name for the template describing the message or header contents of the queue you want to view.

Member
Name of a member of a partitioned data set for the template describing the message or header contents of the queue you want to view.

Copybook/template usage
Indicates if you want to use a template for a logical view of the data.

1 Use the template specified on the panel (or compile the specified copybook into a template, and use that).

2 Use the last template associated with the data set.

3 No logical view is to be used by the function.

4 Create a dynamic template.

Edit template
Indicates if you want to edit the template before use. You edit the template when you need to change format, selection, data create attributes, and reformatting information for output copy templates.

Type (1,2,S)
The type of editing you want to perform.

You can specify one of the following values:

1 Edit the Record Identification Criteria by field first

2 Edit the Record Selection Criteria by field first

S Edit the source definition for a template using the advanced copybook selection facility.

This option is ignored when editing a dynamic template.

Include only selected records
When using a template, only the records selected by template processing are included in the View session.

This option does not alter the operation of view if a template is not used.

Create audit trail
Determines if File Manager generates an audit report of all successful modifications to data made during an Edit session.

The display of this option depends on whether SAF-rule controlled is in effect. See “SAF-rule controlled auditing” on page 47.

The ability to change this option depends on the installation options (in the FMN0POPT macro).

When your system administrator has enforced audit logging, the **Create audit trail** option in this panel is ignored.

When your system administrator has not enforced audit logging, you can set this option on or off for any particular editing task. Type a "/" in the option entry field to generate audit reporting for the current Edit session.

For more information about setting the Audit Trail options, refer to the *File Manager Customization Guide* or see your systems administrator.

Parent panels

- "WebSphere MQ Queue List panel" on page 736

Child panels

- "View panel" on page 707

Related tasks and examples

- "Working with WebSphere MQ" on page 331

WebSphere MQ Queue Information panel

This panel provides the attributes of the queue that are typically displayed as a result of the DISPLAY QLOCAL ALL command, including a terse description of the attribute.

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager            Websphere MQ Queue Information

Queue:
   Queue name . . . . . . . FMN.RFM0713.NEWQUEUE
   Description  . . . . . . RFM0713 TEST QUEUE FOR DATA CREATE TEST

Basic Information:
   Last altered date  . . . 2009-07-28     Time  . 09.04.36
   Creation date  . . . . . 2009-07-20     Time  . 09.30.48

Queue and namelist names:
   Backout requeue name . .
   Cluster namelist . . . .
   Cluster manager  . . . .
   Initiation queue . . . .
   Transmission queue . . .
   Queue manager name . . .
   Remote name  . . . . . .
   Remote manager . . . . .

Maximums:
   Depth  . . . . . . . . . 999999999     Message Length  . . . . 4194304

Cluster:
   Name . . . . . . . . . .
   Available date . . . . .               Available time  . . . .
   Queue type . . . . . . .               Put behaviour . . . . . QMGR
   Workload priority  . . . 0             Workload rank . . . . . 0

Defaults:
   Message binding  . . . . OPEN          Put response  . . . . .
   Priority . . . . . . . . 0             Persistance . . . . . . NO
   Read Ahead . . . . . . . NO            Share option  . . . . . EXCL
   Type . . . . . . . . . . PREDEFINED    Get . . . . . . . . . . ENABLED

Trigger:
   Triggers active  . . . .               Data  . . . . . . . . .
   Depth  . . . . . . . . . 1             Threshold message pri . 0
   Type . . . . . . . . . . FIRST
Other:
   Accounting enabled . . . QMGR          Current depth . . . . . 0
   Backout threshold  . . . 0             Monitoring  . . . . . . QMGR
   Sharing group  . . . . .               Usage . . . . . . . . . NORMAL
   Harden backout . . . . .               Index type  . . . . . . NONE
   Input handles  . . . . . 0             Output handles  . . . . 0
   Delivary sequence  . . . PRIORITY      Reliability level . . . NORMAL
   TPIPE names  . . . . . .
   Process  . . . . . . . .               Property control  . . . COMPAT
   Put enabled  . . . . . . ENABLED       Queue depth High  . . . 80
   Queue depth low  . . . . 40            High depth events . . . DISABLED
   Low depth events . . . . DISABLED      Max depth events  . . . ENABLED
   Service interval . . . . 999999999     Queue type  . . . . . .
   Retention interval . . . 999999999     Share . . . . . . . . .
   Collect stats  . . . . .               Storage class . . . . . DEFAULT
   Target . . . . . . . . .               Target type . . . . . .
Command ===>                                                  Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 285.

Parent panels

- "WebSphere MQ Queue List panel"

Child panels

- None.

Related tasks and examples

- "Working with WebSphere MQ" on page 331

WebSphere MQ Queue List panel

This panel is used to select an item from a list.

You can use it, for example, to select an WebSphere MQ queue name that you want to view.

Panel and field definitions

```
 Process   Options   Help
 ------------------------------------------------------------------------------
 File Manager               Websphere MQ Queue List CSQ2     Row 0001 of 0109

  Queue                           ↓ Descr                                     →
  <---+----10---+----2----+----3--> <---+----10---+----2----+----3----+----4->
_ CICS01.INITQ                      CKTI initiation queue
_ COOL                              Coolest queue ever
_ CSQ1.XMIT.QUEUE                   Transmission queue for CSQ1
_ CSQ2.DEAD.QUEUE                   CSQ2 dead-letter queue
_ CSQ2.DEFXMIT.QUEUE                CSQ2 default transmission queue
_ CSQ3.XMIT.QUEUE                   Transmission queue for CSQ3
_ CSQ4IVPG.TRIGGER                  WebSphere MQ IVP INITIATION QUEUE
_ CSQ4IVP1.TRIGGER                  WebSphere MQ IVP INITIATION QUEUE
_ D3.CSQ2.ANYQ
_ FMN.TEST.NOGET                    RFM0037 GET NOT ENABLED
_ FMN.TEST005                       RFM0037 MQ SERIES TEST Q 1
_ FMN.TEST006                       RFM0037 MQ SERIES TEST Q 1
_ FMN.TEST007                       RFM0037 MQ SERIES TEST Q 1
_ FMN.TEST008                       RFM0037 MQ SERIES TEST Q 1
_ FMN.TEST009                       RFM0037 MQ SERIES TEST Q 1
Command ===> ____________________________________________________ Scroll PAGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=RFind     F7=Backward
 F8=Forward   F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Figure 286.

You can enter any of these prefix commands in the prefix area against a listed queue:

A Add queue. Provide a way to issue a queue definition.

D Delete queue. Provide a way to issue a delete for the queue.

E Edit queue. Start the edit dialog with this queue as the source.

I Information. Provide a way to view queue information in one panel.

R Reset queue. Provide a way to discard all current messages on a queue.

V View queue. Provide a way to display the contents of the queue.

Parent panels

- "WebSphere MQ Managers panel" on page 729

Child panels

- "WebSphere MQ Queue Editor Entry panel" on page 732 (with E prefix command)
- "WebSphere MQ Queue Information panel" on page 734 (with I prefix command)

Related tasks and examples

- "Working with WebSphere MQ" on page 331

Write Tape Mark panel

Panel and field definitions

```
 Process   Options   Help
--------------------------------------------------------------------------------
File Manager                  Write Tape Mark

Tapes:  No tapes allocated


Output:
   DDNAME to use . ________     enter new name, or select one from above
   Tape mode . . . __           optional recording mode or density code
   Tape marks  . . 1___         number of tape marks to be written









Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 287. Write Tape Mark panel

Parent panels

"Tape Specific Functions panel" on page 677

Child panels

Related tasks and examples

Chapter 15. Primary commands

This chapter describes the syntax, parameters and operation of the primary commands that can be used in various File Manager panels. Within each command description, the panels and functions in which the command is available are listed. Where a command has differing functionality on different panels, an explanation of how the command operates in each context is provided.

Primary commands are entered on the Command line of a panel. Some primary commands have equivalent function keys and, where applicable, the default settings for these keys are listed. However, because keylists can be customized through ISPF, your function key settings may differ from those described in this document.

Where the syntax allows you to specify more than one operand, you can use either a blank or a comma as a separator between each operand.

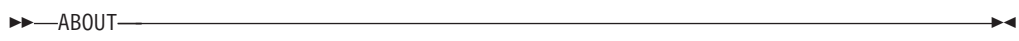
For example, both of the following commands are correct:

```
SORT col1 col2
SORT col1,col2
```

ABOUT primary command

The ABOUT command displays the current File Manager version number and the PTF number of the File Manager "base" component in a window. The popup also shows the copyright information and any notes from IBM that are shipped with the product.

Syntax



Availability

Available on all panels

Related tasks and examples

"Checking your File Manager version" on page 11

AMSMSG primary command

The AMSMSG primary command displays the IDCAMS messages, if any, associated with the current task.

Note: You can also invoke the AMSMSG primary command by selecting "AMS message" from the Process pull-down menu.

Syntax

Availability

- "AIX Association Information panel" on page 446
- "AIX Entry Detail panel" on page 447
- "Catalog Services Data Set List panel" on page 466
- "GDG Entry Detail panel" on page 581
- "Non-VSAM Association Information panel" on page 607
- "Non-VSAM Entry Detail panel" on page 609
- "Path Entry Detail panel" on page 612
- "VSAM Association Information panel" on page 719
- "VSAM Define panel" on page 719
- "VSAM Entry Detail panel" on page 722

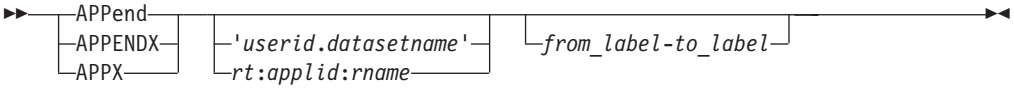
Related tasks and examples

- "Working with a list of catalog entries" on page 314
- "Viewing your Catalog Entry Details" on page 316
- "Defining a new Catalog Entry" on page 319
- "Altering an existing Catalog Entry" on page 322
- "Deleting a catalog entry" on page 323

APPEND, APPENDX primary commands

The APPEND and APPENDX primary commands append to another data set from specified lines of the data in the current editor session.

Syntax



userid.datasetname
Name of fully-qualified data set to which lines of the data are to be appended.

rt:*applid*:*rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
- **FI** For a CICS file.
- **TD** For a Transient Data Queue.
- **TS** For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

from_label
Label indicating first line to be appended to the member or data set.

to_label
Label indicating last line to be appended to the member or data set.

Availability

Available on the edit and view panels. The target data set does not include data on a MQ queue.

Related tasks and examples

- Chapter 6, "Managing data sets," on page 235

ASSOCS primary command

The ASSOCS primary command displays the Association Information panel for the current data type.

Syntax

Availability

- "AIX Entry Detail panel" on page 447
- "Non-VSAM Entry Detail panel" on page 609
- "VSAM Entry Detail panel" on page 722

Related tasks and examples

- "Viewing association information" on page 316

AUTORTRY primary command

The AUTORTRY primary command alters the current setting of the **Auto retry for CHANGE ALL** option.

This setting is usually set by means of the VSAM Edit sharing options from the options menu or options pulldown.

Syntax

Notes:

1 If no parameter is specified: if the current setting is ON, acts as if OFF as been specified; if the current setting is OFF, acts as if ON has been specified.

OFF Disables the **Auto retry for CHANGE ALL** option.

ON Enables the **Auto retry for CHANGE ALL** option.

Availability

- "Editor panel" on page 534

Related tasks and examples

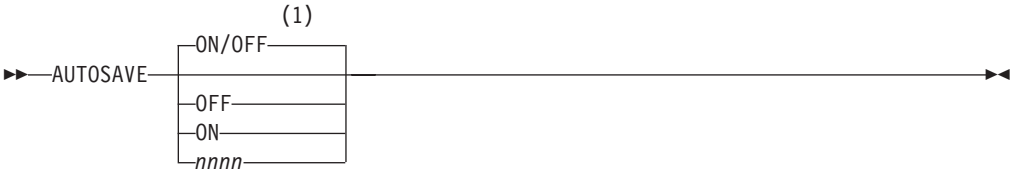
- "Working with File Manager and shared files" on page 89

AUTOSAVE primary command

The AUTOSAVE primary command alters the current setting of the **Autosave frequency** option.

This setting is usually set via the VSAM Edit sharing options from the options menu or options pulldown.

Syntax



Notes:

1 If no parameter is specified: if the current setting is ON (a value exists), acts as if OFF as been specified; if the current setting is OFF, acts as if ON has been specified.

OFF Saves the value of the current **Autosave frequency** and then sets the frequency to zero, disabling the feature.

ON Makes the previously used **Autosave frequency** (or the value 1 if no previous **Autosave frequency** value exists) the current **Autosave frequency**, enabling the feature.

nnnnnn
Sets the current **Autosave frequency** to *nnnnnn*. To disable the feature, specify the value 0. *nnnnnn* can be in the range 0–999999.

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Working with File Manager and shared files" on page 89

BOTTOM primary command

The BOTTOM primary command scrolls to the last page of data. You can use any abbreviation of the BOTTOM command, e.g. BOT or B.

Syntax

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

BOUNDS primary command

The BOUNDS primary command limits the columns searched by the CHANGE, FIND, and EXCLUDE commands. This command is only available in CHAR, HEX or LHEX display formats. It is not available at all when using a segmented record template.

Syntax

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length.

col2 The last column to be included in the range of columns to be searched. If specified, it must be greater than or equal *col1* and less than or equal to the maximum record length. When specified, all columns in the range defined by *col1* to *col2* are searched, inclusively. If not specified, it is assumed to be equal to *col1*, giving a range of one column to be searched.

Note: If neither *col1* nor *col2* are specified, the range is assumed to be from Column 1 to the end of the record. That is, issuing the BOUNDS command with no parameters resets the boundaries to the full record length.

If this command is used in the Find/Change utility and the **JCL source format** option has been selected, the columns searched are set to 3 through 71, unless the statement is not a JCL statement. A statement is considered to be a JCL statement if it begins with the strings "/*" or "//". If the statement does not begin with either of these strings, it is not considered to be a JCL statement in which case any column range specified is honored.

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Limiting the effect of editing changes" on page 108
- "Editing records within data sets" on page 115

CANCEL primary command

The CANCEL primary command (F12) quits the current panel without saving any changes to the data. In an Edit session, if you have entered changes to your data, you are asked to confirm that you want to discard those changes.

Note: Changes to data are only processed when you press Enter or issue a command. If you overtype some data but do not press Enter or issue any other command before you attempt to CANCEL, you are not prompted to confirm the CANCEL action as the data change has not yet been flagged.

QUIT is a synonym for CANCEL.

If the function you were working in was a list in which multiple entries were selected, CANCELing from one of the selected entries halts the processing of other selections from that list.

Syntax

Availability

- Available on all panels.

Related tasks and examples

- "Ending an editor session" on page 60

CAPS primary command

The CAPS or CAPS ON commands instruct File Manager to convert fields or records to uppercase, when a subsequent data change takes place. The change affects either the field in which the change is entered or the entire record, depending upon the current display format:

- In CHAR, HEX or LHEX display format, the entire record is translated to uppercase.
- In SNGL or TABL display format, only the current field is translated to uppercase.

Any CAPS command overrides any previously entered CASE command, and any CASE command overrides any previously entered CAPS command.

Syntax

ON Turns the "convert to uppercase" feature on. Equivalent to issuing the CAPS command without parameters or to issuing the CASE UPPER command.

OFF Turns the "convert to uppercase" feature off, so that subsequent changes to

data do not cause the record or field to be translated to uppercase. Equivalent to the CASE or CASE MIXED commands.

Availability

- "Editor panel" on page 534

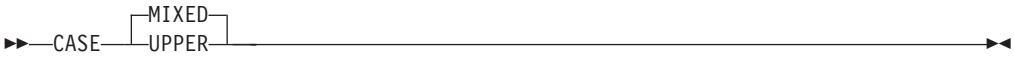
Related tasks and examples

- "Changing the case of data" on page 103

CASE primary command

The CASE command determines whether, when a subsequent data change takes place, fields or records are translated to uppercase (CASE UPPER), or they are left unchanged (CASE MIXED).

Syntax



UPPER
Translates all new records from lowercase to uppercase, and translates all data in changed records from lowercase to uppercase, regardless of how the records are changed. When the data is displayed in CHAR, HEX, or LHEX mode CASE UPPER affects all changed records, including records displayed on the screen which were changed before the CASE UPPER command is entered.

MIXED
Turns the "convert to uppercase" feature off, so that subsequent changes to data do not cause the record or field to be translated to uppercase. Equivalent to issuing the CASE command without a parameter or to issuing the CAPS OFF command.

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Changing the case of data" on page 103

CCSID primary command

In an editor session, the CCSID command toggles on and off the display of the CCSID assigned to each field, if applicable, when using SNGL display format.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

CE (Change Error) primary command

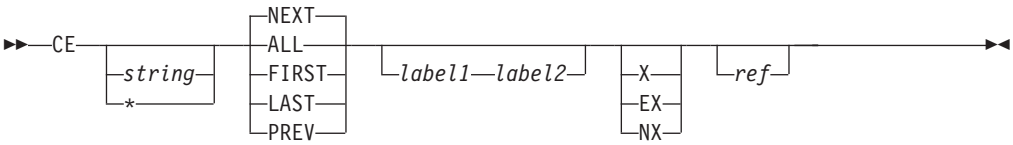
The CE (Change Error) primary command allows you to change numeric fields flagged as being in error (all asterisks in the field display) to a specific numeric value.

The find argument for the CE command is assumed to be a numeric field in error.

The CE command is only valid in SNGL or TABL display formats, when in edit or view.

For details on limiting the columns, fields or records searched by the CE command, see "Limiting the effect of editing changes" on page 108.

Syntax



string A numeric value used to replace the invalid numeric value.

*** (asterisk)**
When used in place of the replacement string, uses the replacement string specified on the previous CE or CHANGE command as the replacement string. CE with no parameters, following a CE or CHANGE with a replacement string, acts the same as `CE *`.

ALL Searches forwards from the top of data. Same as FIRST except that ALL also displays the total number of invalid numeric fields in the records searched.

Note: Suppressed or not-selected records that are hidden from display or represented by shadow lines are not searched, even when you specify ALL.

FIRST Searches from the top of data for the first occurrence of an invalid numeric field.

LAST Searches backwards from the bottom of the data for the last occurrence of an invalid numeric field.

NEXT Searches forwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of an invalid numeric field.

If the next occurrence of the string is in an excluded record and you do not limit the search to not-excluded records, only the first excluded record containing the string is shown.

PREV Searches backwards from the cursor position (if the cursor is within the

data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of an invalid numeric field.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

EX Excluded records only.

NX Not-excluded records only.

X Same as EX.

ref A field reference, specifying the field to be included in the search. For example: #3.

Availability

- "Editor panel" on page 534

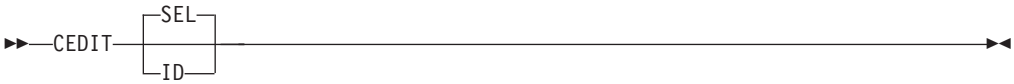
Related tasks and examples

- "Changing invalid numeric fields in your data" on page 108

CEDIT primary command

The CEDIT command displays the template editing panel that is appropriate for the current template type. It is synonymous with the DEDIT command.

Syntax



SEL Default. Displays the Record Selection Criteria panel or the Dynamic Template panel, as appropriate for the current template.

If you are currently using a dynamic template, the CEDIT or CEDIT SEL (or DEDIT) command displays the Dynamic Template panel, where you can select and edit the existing fields in the template, create new fields and enter record selection criteria by field, before returning to viewing the data set with the updated template.

If you are using a template that was generated from a copybook, the CEDIT or CEDIT SEL (or DEDIT) command displays the Record Selection Criteria panel, where you can enter criteria by field, before returning to viewing the data set with the updated template.

ID Displays the Record Identification Criteria panel for the currently selected Record Type in a copybook template. If the copybook contains multiple record types and no record type selection has yet been defined, this is the first record type described in the copybook.

If the current template is dynamic, this parameter is invalid.

If you are not using a template, the CEDIT, CEDIT SEL or CEDIT ID (or DEDIT) command displays the Dynamic Template panel, where you can create and select fields in a new dynamic template and enter record selection criteria by fields.

Availability

- Chapter 3, "Viewing and changing data sets," on page 49

Related tasks and examples

- "Creating dynamic templates" on page 152
- "Editing a template" on page 156

CHANGE/CX primary command

The CHANGE command finds and replaces occurrences of a string or value. It can be entered in an Edit session or when using the Find/Change Utility, by using any abbreviation of the command (for example, C, CH, CHG).

The CX command displays an Extended Command Entry panel, in which you can enter long arguments that may not otherwise fit within the Command line (see "Handling long strings in CHANGE" on page 107).

Note:

1. When you enter the CHANGE primary command in the Editor without specifying any parameters, File Manager displays the Extended Command Entry pop-up panel to allow you to enter arguments.
2. If you use the CNot version of the command, then a string is matched if it does not match according to the matching criteria described in this section.

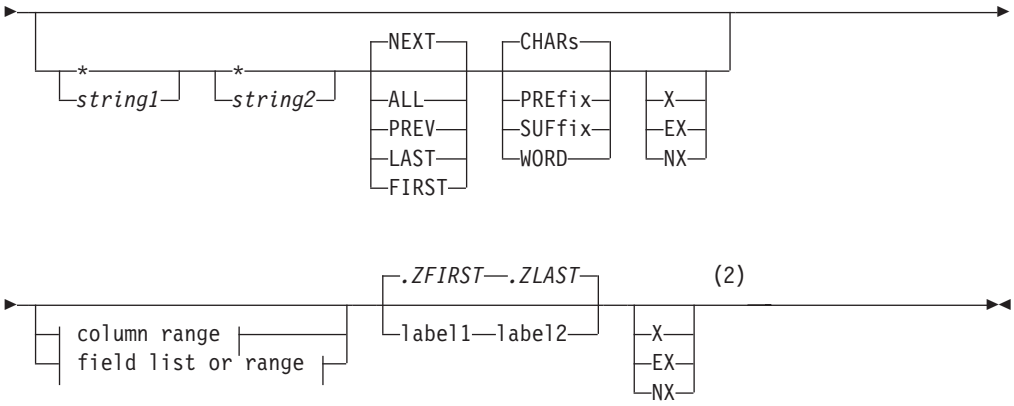
Syntax

CHANGE primary command in the Editor

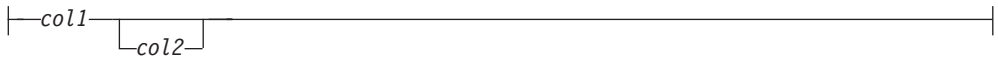
In an Edit session, the CHANGE primary command finds and replaces one or more occurrences of a character string within the currently displayed data set or data set member. In SNGL or TABL display format, you can also use the CHANGE command to find and replace a numeric value in numeric fields.

Note: When you enter the CHANGE primary command in the Editor without specifying any parameters, File Manager displays the Extended Command Entry pop-up panel to allow you to enter arguments.

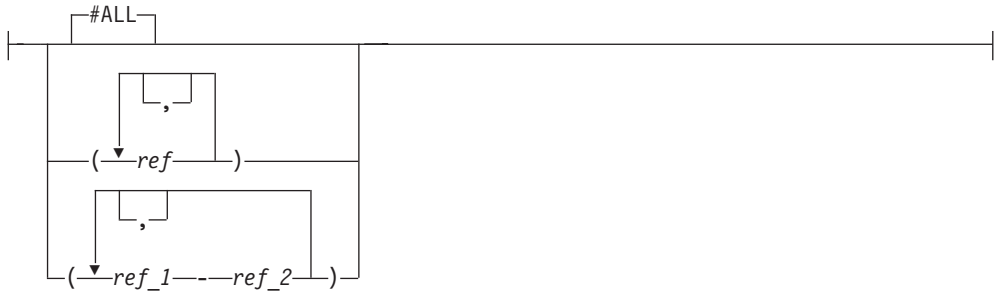
Syntax in the Editor



column range (all display formats):



field list or range (SNGL or TABL display format):



Notes:

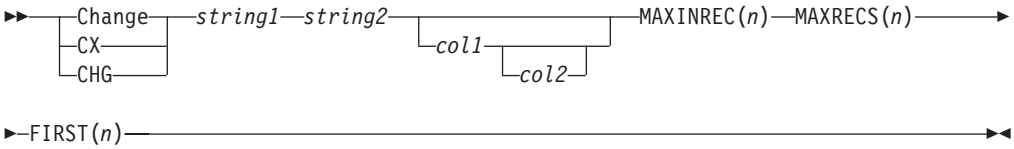
1 You can specify parameters in any order.

2 If none of these parameters (X, EX or NX) is specified, then both excluded and not-excluded records are searched.

CHANGE primary command in the Find/Change Utility

In the FCH utility, the CHANGE primary command finds and replaces all occurrences of the specified string in the selected members or data set.

Syntax in the Find/Change Utility



Change, CX

The CHANGE command or any of its abbreviations is entered on the Command line, together with the desired parameters. To find and replace long strings that do not fit on the Command line, enter the CHANGE

primary command (or one of its abbreviations, such as C) with no parameters, or enter the CX command on the Command line with no parameters. This displays the Extended Command Entry panel, in which you can enter long strings and the CHANGE command parameters.

*** (asterisk)**
When used in place of the search string, uses the search string specified on the previous CHANGE command as the search string.

When used in place of the replacement string, uses the replacement string specified on the previous CHANGE command as the replacement string.

The position of the * is important for CHANGE. If it is positioned before a string, it indicates the previous search argument; that is, it is treated as the first string. If a string is found prior to the *, then it is treated as the second string (the change argument), taking the change argument from the previous CHANGE command. To use both the previous search string and the previous change string, specify `CHANGE * *`.

string1
The string you want to search for, which must be no more than 100 characters long. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, the following command changes the strings `black`, `Black`, and `BLACK`:

  ```
  CHANGE black white
  ```

- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`. The string can be a null string (`''`). If *string1* is a null string, then *string2* is inserted at the current column position.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`. The string can be a null string (`C''`). If *string1* is a null string, then *string2* is inserted at the current column position.
- A DBCS string or a mixed DBCS and non-DBCS string. A shift-out (X'OE') as the first character, or a shift-in (X'OF') as the last character, are discarded as part of the search string.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:
  = Any character.
  @ Alphabetic characters.
  # Numeric characters.
  $ Special characters.
  & notsym;
  Non-blank characters.
  . Invalid characters.
  - Non-numeric characters.
  < Lowercase alphabetics.
  > Uppercase alphabetics.

When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.

- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).
- A numeric value (only when, in SNGL or TABL display format, you limit the search by specifying field references, and only when the field being searched is a numeric field). For details, see "Searching numeric fields" on page 85.

string2

The string that you want to use to replace *string1*, which must be no more than 100 characters long. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. If CAPS ON or CASE UPPER is in effect, then *string2* is translated to uppercase.
- A character string enclosed in quotation marks. The string can contain blanks and commas. If CAPS ON or CASE UPPER is in effect, then *string2* is translated to uppercase. The string can be a null string (`''`).
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. Case is respected and retained. The string can be a null string (`C''`).
- A DBCS or a mixed DBCS and non-DBCS string. The shift-out (X'0E') and shift-in (X'0F') at the start and end of the string are adjusted in such a way that the integrity of the DBCS data is not compromised.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe the change to be made. You can change characters from uppercase to lowercase or from lowercase to uppercase, or leave the field the same using these string special characters.

  = Any character.
  < Lowercase alphabetics.
  > Uppercase alphabetics.

  Examples of picture strings used with the CHANGE command:

  `c p'.' x'00'`
  Change next invalid character to hex x'00'.

  `c p'###' 100`
  Change next 3-digit number to 100.

  `c all 73 80 p'='  " "`
  Put blanks in columns 73 to 80.

  `c all p'-' 1 10 "0"`
  Change all non-numeric characters in columns 1 to 10 into the char "0".

  `c p'<' p'>'`
  Change any lowercase to uppercase.

  `c p'a>' p'=<'`
  Change char a, any uppercase to char a, lowercase.

  `c p'>' '¬'`
  Change any uppercase to "not sign" (note: "not sign" has no special meaning in a "to" string).

c p'<#' p'>='
Change any lowercase, any number to uppercase, the number found.

When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process. .

- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by X ('C1C2'X). Case is respected and retained (the hexadecimal values are used exactly as specified).
- A numeric value (SNGL and TABL display formats only).

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length. When column ranges are specified and the data is displayed in SNGL or TABL formats, the search is performed in the order that the data occurs, which may not be the order in which the fields are displayed.

When column ranges are specified in SNGL or TABL display format, then a string comparison is performed, even for numeric fields.

Note: When you use the CHANGE primary command in the Find/Change Utility, if the **JCL Source format** option is selected, the columns searched are set to 3 through 71, unless the statement is not a JCL statement. A statement is considered to be a JCL statement if it begins with the strings "/*" or "//". If the statement does not begin with either of these strings, it is not considered to be a JCL statement in which case any column range specified on the FIND (or CHANGE, respectively) command or preset using the BOUNDS command is honored. If no column range has been specified, the full record is searched.

ALL (Applies in the Editor only.) Causes the search to begin at the top of the data and find and replace all occurrences of the string.

Note:

1. Suppressed or not-selected records that are hidden from display or represented by shadow lines are not processed by the CHANGE command, even when the ALL parameter is specified.
2. The ALL parameter is not recursive when used with the CHANGE command. That is, applying the command CHANGE ALL 'SS' 'RS' to the string 'SSS' results in 'RSS', not in 'RRS'.

FIRST (Applies in the Editor only.) Searches from the top of data for the first occurrence of *string*.

LAST (Applies in the Editor only.) Searches backwards from the bottom of data for the last occurrence of *string*.

NEXT (Applies in the Editor only.) Causes the search to begin at the cursor location (if the cursor is within the data portion of the screen) or the beginning of the first record displayed, and searches ahead to find the next occurrence of the string. If the next occurrence of the string is in an excluded record and you do not limit the search to not-excluded records, only the first excluded record containing the string is shown.

PREV (Applies in the Editor only.) Searches backwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of *string*.

CHARS
Matches the search string anywhere in the data.

PREFIX
Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX
Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD
Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length. If not specified, the string is matched with data starting in the *col1* location for the length of the string.

#ALL Each field is searched according to its template attributes.

ref (Applies in the Editor only.) A field reference, specifying the field to be included in the search, for example: `#3`. When the field is an item in an array, you must specify a subscript in parentheses to identify the occurrence that you want to change, for example: `#5(3)`. If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array, for example: `#7(2,3)`.

Multiple field references must either be enclosed in parentheses, or separated with commas but without any intervening spaces. For details, see "Limiting the search to specified fields" on page 111.

In SNGL or TABL display format, if you specify field references or do not specify a field reference or column range, when searching a numeric field, the search string is interpreted as a numeric value, and a numeric comparison is performed. When searching a character field, a string comparison is performed. That is, the search is performed based on the File Manager template attributes. For details, see "Searching numeric fields" on page 85.

ref_1 The first field reference of a range of fields. It cannot be subscripted. If the *ref_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

ref_2 The last field reference of a range of fields. It cannot be subscripted. If the *ref_2* field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *ref_1* and *ref_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *ref_1* is a higher value than *ref_2*, the search process reverses the operands.

EX (Applies in the Editor only.) Excluded records only.

label1 (Applies in the Editor only.) Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 (Applies in the Editor only.) Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

NX (Applies in the Editor only.) not-excluded records only.

X (Applies in the Editor only.) Same as EX.

MAXINREC(*n*)
(Applies in the FCH Utility only.) The MAXINREC parameter allows you to specify the maximum number of **changes** that can be made within a **single record**.

MAXRECS(*n*)
(Applies in the FCH Utility only.) The MAXRECS parameter allows you to specify the maximum number of **records** that can be changed within a **single data set or PDS member**.

FIRST(*n*)
(Applies in the FCH Utility only.) The FIRST parameter allows you to specify the maximum number of **total changes** that can be performed within a **single data set or PDS member**.

Availability

- "Editor panel" on page 534
- "Find/Change Utility panel" on page 574

Related tasks and examples

- "Finding and replacing strings" on page 104
- "Finding and changing data in multiple PDS members" on page 270
- "Handling long strings in CHANGE" on page 107
- "Using File Manager functions in batch jobs" on page 401
- "Working with files or members containing JCL" on page 279

CHGASAVE primary command

The CHGASAVE primary command alters the current setting of the **CHANGE ALL autosave frequency** option.

This setting is usually set by means of the VSAM Edit sharing options from the options menu or options pulldown.

Syntax

Notes:

1 If no parameter is specified: if the current setting is ON (a value exists), acts as if OFF as been specified; if the current setting is OFF, acts as if ON has been specified.

OFF Saves the value of the current **CHANGE ALL autosave frequency** and then sets the frequency to zero, disabling the feature.

ON Makes the previously used **CHANGE ALL autosave frequency** (or the value 1 if no previous **CHANGE ALL autosave frequency** value exists) the current **CHANGE ALL autosave frequency**, enabling the feature.

nnnnnn
Sets the current **CHANGE ALL autosave frequency** to *nnnnnn*. To disable the feature, specify the value 0. *nnnnnn* can be in the range 0–999999.

Availability

- "Editor panel" on page 534

Related tasks and examples

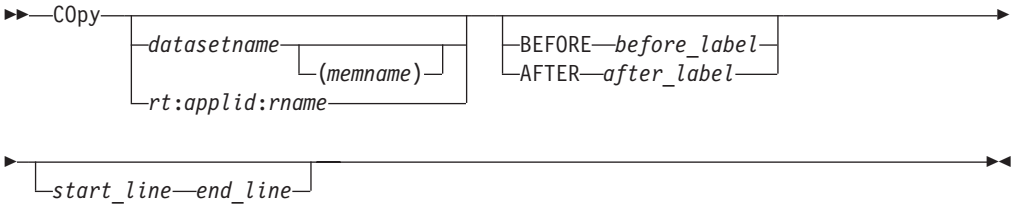
- "Working with File Manager and shared files" on page 89

COPY primary command

The COPY command copies one or more lines of data from a data set or member of a partitioned data set (PDS) into the member or data set currently being edited or viewed. Use the A (after) or B (before) line commands or the AFTER or BEFORE keyword along with a label, to specify where the data is to be copied.

You can optionally specify the data set details from which the data is to be copied. If you do not specify these details, File Manager displays the Edit/View - Copy panel where you can enter the data set details from which the data is to be copied.

Syntax



datasetname
Name of data set from which data is to be copied. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

rt:*applid*:*rname*
: You can specify a CICS resource in place of a data set name, where:

  rt Resource type. Valid values are:
  - **FI** For a CICS file.
  - **TD** For a Transient Data Queue.
  - **TS** For a Temporary Storage Queue.

  applid The VTAM applid of the CICS system.

  rname The name of the resource.

memname
: Name of member from which data is to be copied.

before_label
: Label identifying the start of a range of records from which data is to be copied. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

after_label
: Label identifying the end of a range of records from which data is to be copied. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

start_line
: Number identifying the line number of the first line of data in the data set or member from which data is to be copied.

end_line
: Number identifying the line number of the last line of data in the data set or member from which data is to be copied.

Availability

- "Browse Entry panel" on page 459
- "Editor panel" on page 534

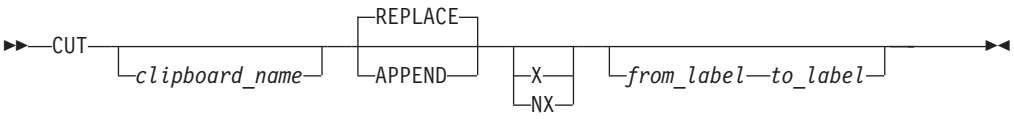
Related tasks and examples

- "Copying data from another data set" on page 122

CUT primary command

The CUT command copies one or more lines of data from the current edit or view session into a clipboard for later retrieval by the PASTE command.

Syntax



clipname
: Name of a clipboard to which data is to be copied. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

APPEND
Copied data is added to the existing contents of the clipboard.

REPLACE
Copied data replaces the existing contents of the clipboard.

NX Copy only not-excluded records.

X Copy only excluded records.

from_label
Label identifying the start of a range of records from which data is to be copied to the clipboard. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

to_label
Label identifying the end of a range of records from which data is to be copied to the clipboard. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

Availability

- "Browse Entry panel" on page 459
- "Editor panel" on page 534

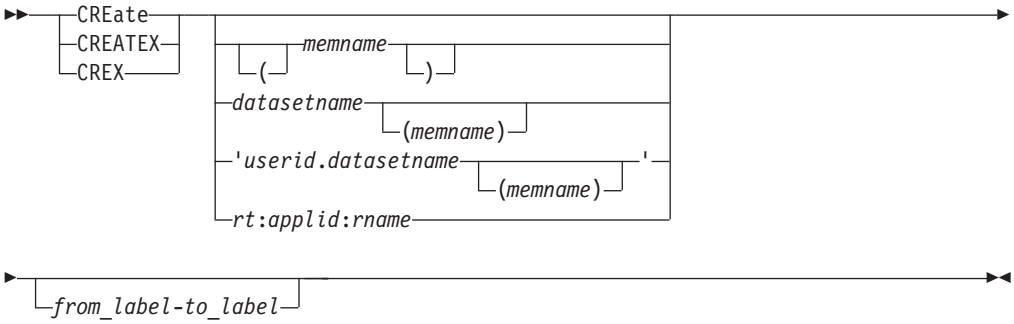
Related tasks and examples

- "Copying data to and from a clipboard" on page 121

CREATE, CREATEX primary commands

The CREATE and CREATEX primary commands create another member or data set from specified lines of the data in the current editor session.

Syntax



memname
Name of member to be created.

datasetname
Name of data set to be created. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

userid.datasetname
Name of fully-qualified data set to be created.

rt:*applid*:*rname*
: You can specify a CICS Transient Data Queue or CICS Temporary Storage Queue in place of a data set name, where:

  rt Resource type. Valid values are:
  TD For a Transient Data Queue.
  TS For a Temporary Storage Queue.

  applid The VTAM applid of the CICS system.

  rname The name of the resource.

from_label
: Label indicating first line to be copied into the new member or data set.

to_label
: Label indicating last line to be copied into the new member or data set.

Availability

Available on the edit and view panels.

Related tasks and examples

- Chapter 6, "Managing data sets," on page 235

DEDIT primary command

The DEDIT command displays the template editing panel that is appropriate for the current template type. It is synonymous with the CEDIT SEL command.

Syntax

If you are currently using a dynamic template, the DEDIT (or CEDIT or CEDIT SEL) command displays the Dynamic Template panel, where you can select and edit the existing fields in the template, create new fields and enter record selection criteria by field, before returning to viewing the data set with the updated template.

If you are using a template that was generated from a copybook, the DEDIT (or CEDIT or CEDIT SEL) command displays the Record Selection Criteria panel, where you can enter criteria by field, before returning to viewing the data set with the updated template.

If you are not using a template, the DEDIT command displays the Dynamic Template panel, where you can create and select fields in a new dynamic template and enter record selection criteria by fields.

You can abbreviate the command DEDIT to DE.

Availability

- Chapter 3, "Viewing and changing data sets," on page 49

Related tasks and examples

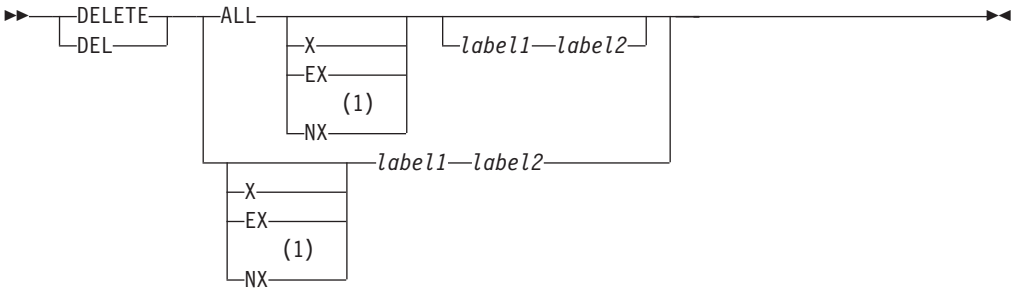
- "Creating dynamic templates" on page 152
- "Editing a template" on page 156

DELETE primary command

The DELETE primary command deletes records from a data set or data set member.

Syntax

Syntax (CHAR, HEX, LHEX or TABL display format)



Notes:

1 If none of these parameters (X, EX or NX) is specified, then the DELETE command applies to both excluded and not-excluded records.

In SNGL display format, DELETE deletes the current record.

Syntax (SNGL display format)

ALL Causes the DELETE command to delete all records (or all excluded or not-excluded records) within the specified range of records (or within the entire data set, if you do not specify a range).

If you do not specify the ALL parameter, then the DELETE command deletes only the first record (or only the first excluded or not-excluded record) within the specified range.

Note: Suppressed or not-selected records that are hidden from display or represented by shadow lines are not affected by the DELETE command, even when the ALL parameter is specified.

EX Excluded records only.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a

period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

NX Not-excluded records only.

X Same as EX.

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Deleting records" on page 116

DESCRIBE primary command

Shows information about a template, including the member name of the source copybook, the name of the template data set (and member name, if a PDS), and the description of the template. You can update the description text.

Syntax

Availability

- "Record Type Selection panel" on page 636
- "Field Selection/Edit panel" on page 565

Related tasks and examples

DOWN primary command

The DOWN primary command scrolls down (forward) through your data. The amount (number of lines) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax

nnnn Scroll down *nnnn* lines.

Csr Scroll down to the cursor position.

Data Scroll down one line less than a page of data.

Half Scroll down half a page of data.

Max Scroll to bottom of file. This has the same effect as the BOTTOM command.

Page Scroll down one page of data.

Scroll field
Scroll down by the amount indicated in the **Scroll field**. This is the default if no parameter is used.

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

DX primary command

The DX command converts a decimal value to its equivalent hexadecimal value.

Syntax

Availability

This command can be entered on any File Manager panel.

Related tasks and examples

"Using hexadecimal values in File Manager" on page 37

EDIT primary command

The EDIT primary command edits the template built from the specified copybooks. If changes have been made, a compile and update is performed.

Syntax

Availability

- "Copybook Selection panel" on page 505

Related tasks and examples

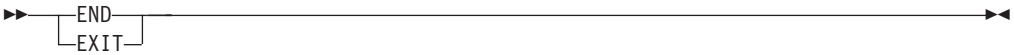
- "Advanced copybook selection" on page 147

END primary command

The END primary command exits from the current panel and stores any editing changes made to data in memory and any changes made to panel values (such as the names of data sets or selected options) in your ISPF profile. Where exiting the current panel completes the editing of a data set, the END command also saves the changes to file. For example, using the END command in the Edit panel saves any changes to file; using END on a "nested" template editing panel, such as the Field Attributes panel, stores changes in memory (these changes are not written to file until you end the entire template editing session).

In an Edit session, the END command is synonymous with the EXIT and FILE commands.

Syntax



Availability

Available on all panels.

Related tasks and examples

- "Ending an editor session" on page 60

EXCLUDE/XX primary command

The EXCLUDE primary command excludes records from display in an editor session. (Not available in SNGL display format.)

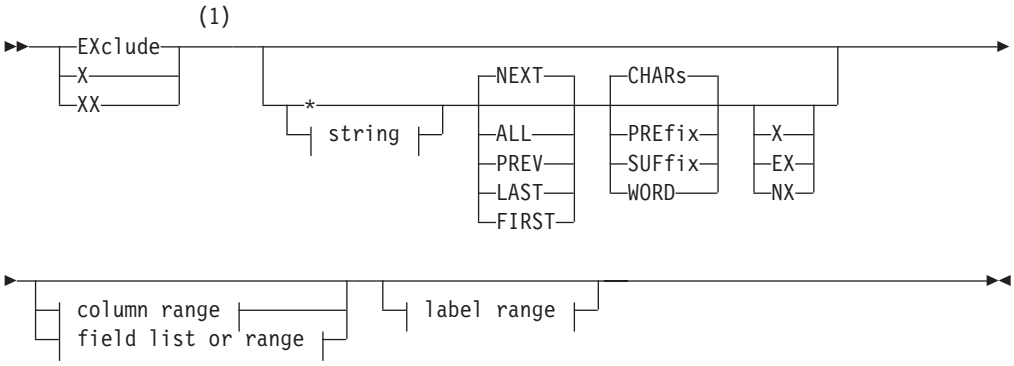
The XX command displays an Extended Command Entry panel, in which you can enter long arguments that may not otherwise fit within the Command line.

EXCLUDE commands are cumulative; each successive EXCLUDE command increases the number of records already excluded.

To "unexclude" (redisplay) excluded records, use the RESET EXCLUDED primary command (see "RESET primary command" on page 806) or the F or LA prefix commands (see "Editor panel" on page 534). If you used a search string to exclude the records then, when you unexclude the records, the search string is highlighted; to remove this highlighting, enter the RESET FIND command.

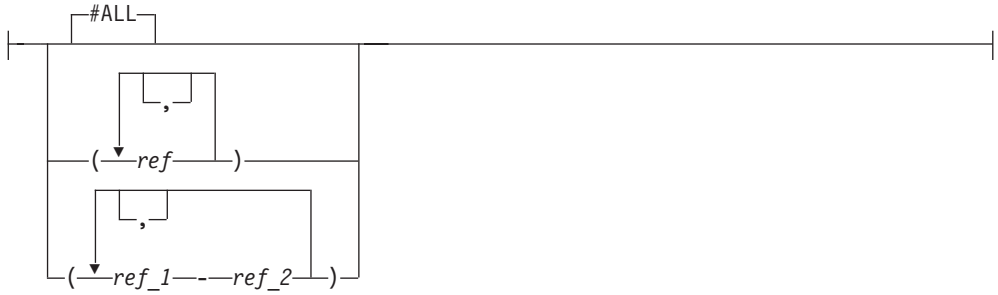
Note: When you enter the EXCLUDE primary command in the Editor without specifying any parameters, File Manager displays the Extended Command Entry pop-up panel to allow you to enter arguments.

Syntax

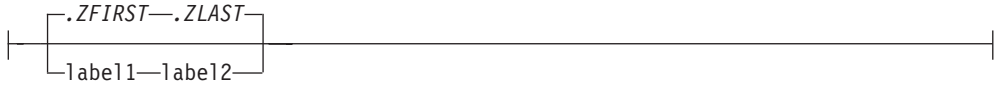


column range (all display formats):

field list or range (SNGL or TABL display format):



label range (Edit only):



Notes:

1 You can specify parameters in any order.

*** (asterisk)**
Uses the search string specified on the previous EXCLUDE command as the search string.

string The search string you want to search for. Records containing this string, within the limits imposed by the other EXCLUDE command parameters, are excluded. The string can be:

- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, `Mixed` matches `MIXED`.
- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.

- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:

  = Any character.
  @ Alphabetic characters.
  # Numeric characters.
  $ Special characters.
  & notsym;
  Non-blank characters.
  . Invalid characters.
  - Non-numeric characters.
  < Lowercase alphabetics.
  > Uppercase alphabetics.

  Examples of picture strings used with the EXCLUDE command:

  ex p'.' 73 80
  Exclude invalid characters in columns 73 to 80.

  ex p'###'
  Exclude 3-digit number (for example, 101 but not 99).

  ex '@1'p 1
  Exclude label a1,b1,c1, (and so on) in column 1.

  ex p'<'
  Exclude the next lowercase alphabetic character.

  ex p'¬' 72
  Exclude the next non-blank character in column 72.

  When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).
- A numeric value (only when, in SNGL or TABL display format, you limit the search by specifying field references, and only when the field being searched is a numeric field). For details, see "Searching numeric fields" on page 85.

ALL Searches all records from the top of data to the bottom.

Note:

1. The EXCLUDE command does not affect suppressed or not-selected records that are hidden from display or represented by shadow lines, even when you specify the ALL parameter.
2. The command EXCLUDE ALL (with no other parameters) excludes all displayed records.

FIRST Searches from the top of data for the first occurrence of *string*.

LAST Searches backwards from the bottom of data for the last occurrence of *string*.

NEXT Searches forwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of *string* in a record that is not already excluded.

PREV Searches backwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of *string*.

PREFIX
Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX
Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD
Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length. When column ranges are specified and the data is displayed in SNGL or TABL formats, the search is performed in the order that the data occurs, which may not be the order in which the fields are displayed.

When column ranges are specified in SNGL or TABL display format, then a string comparison is performed, even for numeric fields.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length. If not specified, the string is matched with data starting in the *col1* location for the length of the string.

#ALL Each field is searched according to its template attributes.

ref A field reference, specifying the field to be included in the search, for example: `#3`. When the field is an item in an array, you must specify a subscript in parentheses to identify the occurrence that you want to use, for example: `#5(3)`. If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array, for example: `#7(2,3)`.

Multiple field references must either be enclosed in parentheses, or separated with commas but without any intervening spaces. For details, see “Limiting the search to specified fields” on page 111.

In SNGL or TABL display format, if you specify field references or do not specify a field reference or column range, when searching a numeric field, the search string is interpreted as a numeric value, and a numeric comparison is performed. When searching a character field, a string

comparison is performed. That is, the search is performed based on the File Manager template attributes. For details, see "Searching numeric fields" on page 85.

ref_1 The first field reference of a range of fields. It cannot be subscripted. If the *ref_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

ref_2 The last field reference of a range of fields. It cannot be subscripted. If the *ref_2* field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *ref_1* and *ref_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *ref_1* is a higher value than *ref_2*, the search process reverses the operands.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Results

Related tasks and examples

- "Limiting the effect of editing changes" on page 108

EXIT primary command

The EXIT primary command is synonymous with the END command. See "END primary command" on page 762 for details.

EXPAND primary command

The EXPAND primary command displays a panel for inputting or viewing long data. You can use this command from any File Manager panel that has a scrollable input or display field (you must position the cursor on the scrollable field).

Syntax

Availability

- Any File Manager panel with a scrollable input or display field.

Related tasks and examples

- "Scrollable input and display fields for long names" on page 80

EXTENT primary command

The EXTENT primary command displays the Non-VSAM Extent Information panel showing extent details for a non-VSAM data set.

Syntax

Availability

- "Non-VSAM Entry Detail panel" on page 609
- "Non-VSAM Extent Information panel" on page 610

Related tasks and examples

- "Displaying extent details for non-VSAM data sets" on page 318

FE (Find Error) primary command

The FE (Find Error) primary command finds the next "field in error". (SNGL and TABL display format only.)

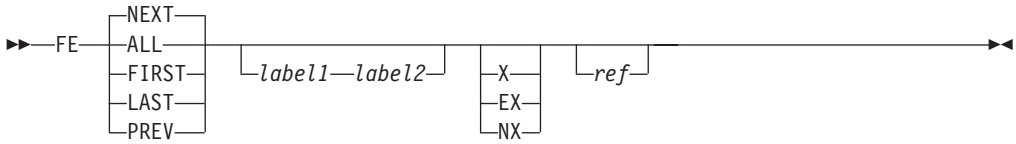
- Numeric fields whose contents cannot be formatted as a valid numeric value
- Fields that run over the record length

File Manager positions the cursor at the beginning of the field and, if necessary, automatically scrolls the data to bring the field into view.

To move the cursor to the next field in error, either enter the FE command again, or press the RFind function key (F5).

For details on limiting the columns, fields or records searched by the FE command, see "Limiting the effect of editing changes" on page 108.

Syntax



ALL Searches forwards from the top of data. Same as FIRST except that ALL also displays the total number of fields in error in the records searched.

Note: Suppressed or not-selected records that are hidden from display or represented by shadow lines are not searched, even when you specify ALL.

FIRST Searches from the top of data for the first occurrence of an invalid value.

LAST Searches backwards from the bottom of the data for the last occurrence of an invalid value.

NEXT Searches forwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of an invalid value.

PREV Searches backwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of an invalid value.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

EX Excluded records only.

NX Not-excluded records only.

X Same as EX.

ref A field reference, specifying the field to be included in the search. For example: #3.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Finding and correcting errors in your data" on page 86

FKEY primary command

You use the FKEY primary command to specify a starting position when browsing, viewing, or editing a KSDS using a layout from the currently loaded template. The FKEY primary command positions to the record containing, or beginning with, the specified keyvalue. This is similar to the ability provided by the Key field on the editor entry panels.

Syntax

layout_name
If you leave this blank, the layout of the current record is used, or the current key segment layout for segmented editing. If the current record is the top or end of data, then the current layout for the editor is used. You can also specify a value which is used as a filter to select a layout name. If the value does not match any existing layouts or matches more than one layout, then a list is displayed from which you can select the required layout.

Examples

Formatted key showing all current 01s in a selection list:

```
FKEY *
```

Formatted key using default layout:

```
FKEY
```

Formatted key showing all layouts beginning with A:

```
FKEY A
```

Formatted key using layout rec-type01:

```
FKEY rec-type01
```

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "View panel" on page 707

FILE primary command

The FILE primary command issues the SAVE command, writing all changes made in panels used within the current editing task, and then returns you to your data set Entry panel. For example, using the FILE command on a "nested" template editing panel, such as the Field Attributes panel, saves all changes made on that panel, and any changes made on other panels used in the current template editing session, then return you to your data set Entry panel.

In an Edit session, the FILE command is synonymous with the END and EXIT commands.

Syntax

Availability

Available on all panels.

Related tasks and examples

- "Ending an editor session" on page 60

FIND/FX primary command

The FIND command finds occurrences of a string or value.

The FIND command can be entered using any abbreviation of the command (for example, F or FI), or by entering the forward slash ("/") character.

The FX command (only available in View, Edit and the Find/Change Utility) displays an Extended Command Entry panel, in which you can enter long arguments that may not otherwise fit within the Command line (see "Handling long strings in FIND" on page 85 for more information).

Multiple (& or AND) and (| or OR) operations are evaluated strictly left to right, but AND and OR may not be mixed in the same FIND statement. There is no operator precedence, and parentheses cannot be used to change the order of evaluation.

Syntax

FIND command in View or Edit

In View and Edit, the FIND primary command searches the data for a character string or (SNGL or TABL display format only) a numeric field with a particular numeric value. If the string or numeric value is found, then the FIND command positions the cursor at the beginning of the string or numeric field and, if necessary, automatically scrolls the found data into view.

The FIND command highlights all occurrences of the search string or numeric value in the entire data set. To turn off the highlighting, enter the RESET FIND command.

To find the next occurrence of the same string, press the RFind function key (F5), enter FIND *, or enter the FIND command with no parameters. A message is displayed if the string cannot be found.

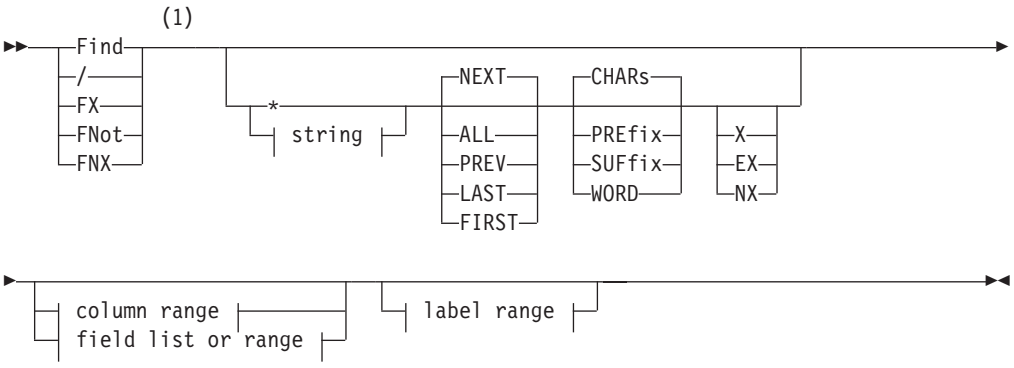
Note:

1. When you enter the FIND primary command in the Editor without specifying any parameters, File Manager displays the Extended Command Entry pop-up panel to allow you to enter arguments.
2. `FIND *` does *not* repeat the previous FIND command with the same parameters. `FIND *` repeats the previous FIND command with the same string argument, but all other parameters revert to their default values unless specified.
3. If you use the FNot version of the command, then a string is matched if it does not match according to the matching criteria described in this section.

For details on limiting the columns, fields or records searched by the FIND command, see “Limiting the effect of editing changes” on page 108.

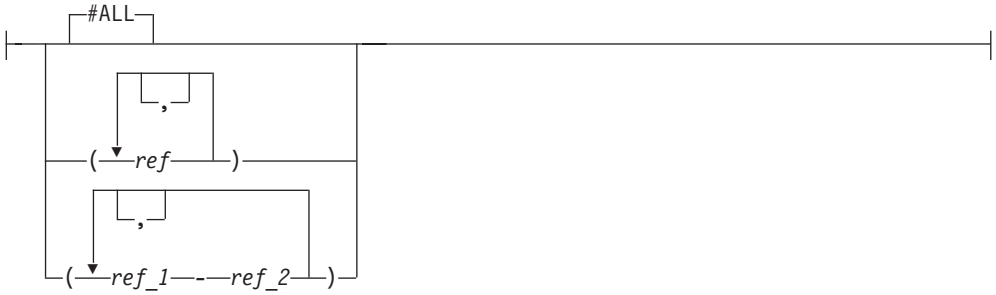
When you are using View or Edit, the syntax of the FIND primary command is:

Syntax in View or Edit



column range (all display formats):

field list or range (SNGL or TABL display format):



label range (Edit only):

Notes:

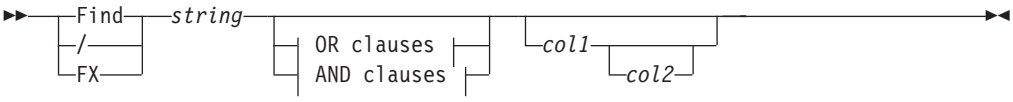
1 You can specify parameters in any order.

FIND command in FCH Utility

In the Find/Change Utility, the FIND command searches contents of the selected data set or data set members. If a matching string is found, the results are displayed on screen in a report showing the member names and record numbers where the string was found, and the list of selected members is reduced to only those containing a match.

When you are using the Find/Change Utility, the syntax of the command is:

Syntax in the Find/Change Utility



OR clauses:

AND clauses:

Related topics
"Using File Manager functions in batch jobs" on page 401

FIND command in Catalog Services: Data Set List, Display VTOC Data Set List, AFP Browse and Memory Browse

In the Catalog Services: Data Set List or Display VTOC Data Set List panels, the FIND command searches the list of data set names. If an entry name containing a matching string is found, the list is scrolled to display the matching data set at the top position. The Display VTOC Data Set List panel allows for a string argument which may contain the * and % wildcard characters. The FIND argument is matched against the data set names using the default system match rules for such wildcard characters.

In the AFP Browse utility, the FIND primary command finds the next occurrence of a character string in the data you browse. If the character string is found in a data line, it is displayed at the top line and the cursor is positioned at the beginning of the string.

In the Memory Browse utility, the FIND primary command finds the next occurrence of a character string in the data you are browsing. If the character string is found, it is displayed at the top position. The **Location field** indicates the address. To find the next occurrence of the same string, use the RFIND command, or enter the FIND command with no argument. A message is displayed if the string cannot be found.

In the Catalog Services: Data Set List, Display VTOC, AFP Browse and Memory Browse panels, the syntax of the command is:

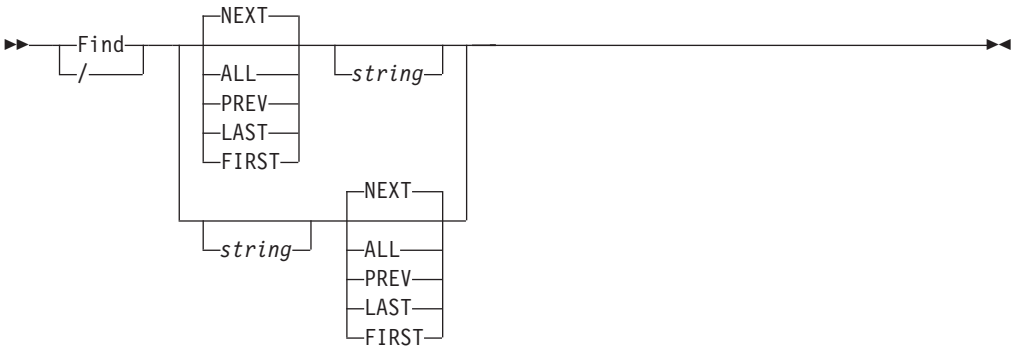
Syntax

FIND command in member selection panels and the Load Module Information panel

In member selection panels, the FIND command normally finds the next occurrence of a character string in the member names being displayed, starting at the top of the displayed page or starting at the cursor position (if the cursor is within the data portion of the display). In the Load Module Information panel, the same rules apply to the list of entries shown on the display.

File Manager places the cursor at the beginning of the found string. If necessary, automatic scrolling is performed to bring the string into view.

Syntax - member selection and Load Module Information panels

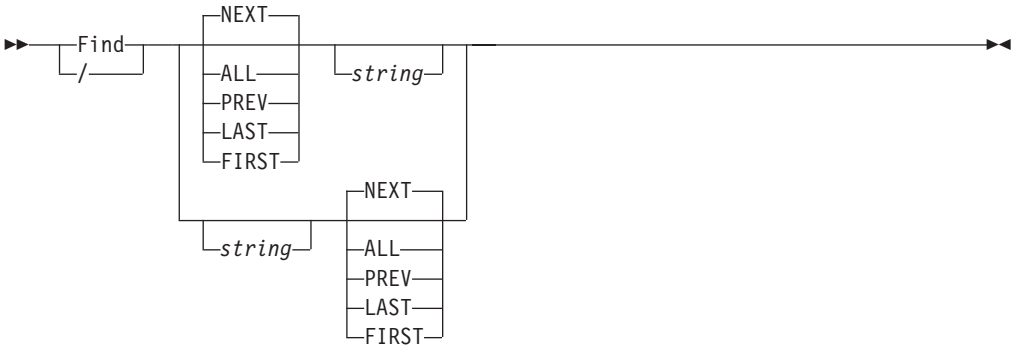


FIND command in WebSphere MQ selection lists

In WebSphere MQ selection lists, the FIND command finds the next occurrence of a character string in any column of the selection list being displayed, starting at the top of the displayed page or at the cursor position (if the cursor is within the data portion of the display).

The cursor is placed at the line containg the found string. If necessary, automatic scrolling is performed to bring the string into view.

Syntax - WebSphere MQ selection lists



*** (asterisk)**
Uses the search string specified on the previous FIND command as the search string.

string Search string, which must be no more than 100 characters long, or when used with,a Member Selection List, no more than the width of the member name column. The string can be:

- No more than 100 characters long.
- An empty string, that is, using FIND without specifying a string. The search string specified on the previous FIND, CHANGE, or EXCLUDE command is used as the search string. A FIND ALL with no search string specified performs a RESET FIND, turning off any highlighting from previous FIND, CHANGE, or EXCLUDE commands. Does not apply to member or WebSphere MQ selection lists.
- A character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is

ignored. Uppercase and lowercase representations of the same character match. For example, the following command finds the strings `black`, `Black`, and `BLACK`:

```
FIND black
```

- A character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored. For example, `'Exact string'` matches `'exact string'`.
- C followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case). For example, `C'Exact string'` does not match `C'exact string'`.
- A DBCS string or a mixed DBCS and non-DBCS string. A shift-out (X'OE')as the first character, or a shift-in (X'OF') as the last character, are discarded as part of the search string. (Not supported in the Catalog Services List or Display VTOC panels.) Does not apply to member or WebSphere MQ selection lists.
- P preceded or followed by a picture string enclosed in single or double quotation marks to describe a type of string to be found rather than the exact characters. It can contain blanks, alphabetic and numeric characters which represent themselves, or any of the special characters listed here, each of which represents a class of characters:

  = Any character.
  @ Alphabetic characters.
  # Numeric characters.
  $ Special characters.
  & notsym;
  Non-blank characters.
  . Invalid characters.
  - Non-numeric characters.
  < Lowercase alphabetics.
  > Uppercase alphabetics.

  Examples of picture strings used with the FIND command:

  find p'.' 73 80
  Find invalid characters in columns 73 to 80.

  find p'###'
  Find 3-digit number (for example, 101 but not 99).

  find '@1'p 1
  Find label a1,b1,c1, (and so on) in column 1.

  find p'<'
  Find lowercase alphabetic character.

  find p'¬' 72
  Find non-blank character in column 72.

  When this notation is used, numeric, bit and unicode fields (for SNGL and TABL display formats) are excluded from the search process.
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X). Does not apply to member selection lists.
- A numeric value (only when, in SNGL or TABL display format, you limit the search by specifying field references, and only when the field

being searched is a numeric field). For details, see "Searching numeric fields" on page 85. Does not apply to member or WebSphere MQ selection lists.

ALL Searches forwards from the top of data. Same as FIRST, except that ALL also displays the total number of occurrences of *string* in the records searched.

Note:

1. Suppressed or not-selected records that are hidden from display or represented by shadow lines are not searched, even when you specify ALL.
2. The ALL parameter is recursive when used with the FIND command. That is, applying the command FIND ALL 'SS' to the string 'SSS' finds two occurrences, the first 'SS' and the second 'SS'.
3. A FIND ALL command with no search string specified performs a RESET FIND command, turning off any highlighting from previous FIND, CHANGE, or EXCLUDE commands.

EX Excluded records only.

FIRST Searches from the top of data for the first occurrence of *string*.

LAST Searches backwards from the bottom of data for the last occurrence of *string*.

label1 Label identifying the start of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

label2 Label identifying the end of a range of records. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

NEXT Searches forwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of *string*.

NX Not-excluded records only.

PREV Searches backwards from the cursor position (if the cursor is within the data portion of the screen), or from the beginning of the topmost record displayed on the screen, for the next occurrence of *string*.

PREFIX
Matches the search string wherever it appears as a prefix in the data. To be a prefix, the matched text must be preceded by a non-alphanumeric character or be the start of a line or field, and must be followed by an alphanumeric character.

SUFFIX
Matches the search string wherever it appears as a suffix in the data. To be a suffix, the matched text must be preceded by an alphanumeric character, and must be followed by a non-alphanumeric character or be the end of a line or field.

WORD
Matches the search string wherever it appears as a word in the data. To be a word, the matched text must be preceded by a non-alphanumeric

character or be the start of a line or field, and must be followed by a non-alphanumeric character or be the end of a line or field.

col1 The first column to be included in the range of columns to be searched. Must be greater than or equal to 1, and less than or equal to the maximum record length. When column ranges are specified and the data is displayed in SNGL or TABL formats, the search is performed in the order that the data occurs, which may not be the order in which the fields are displayed.

When column ranges are specified in SNGL or TABL display format, then a string comparison is performed, even for numeric fields.

Note: When you use the FIND primary command in the Find/Change Utility, if the **JCL Source format** option is selected, the columns searched are set to 3 through 71, unless the statement is not a JCL statement. A statement is considered to be a JCL statement if it begins with the strings "/*" or "//". If the statement does not begin with either of these strings, it is not considered to be a JCL statement in which case any column range specified on the FIND (or CHANGE, respectively) command or preset using the BOUNDS command is honored. If no column range has been specified, the full record is searched.

col2 The last column to be included in the range of columns to be searched. Must be greater than or equal to *col1* and less than or equal to the maximum record length.

In the Editor, if *col2* is not specified, it defaults to the value of *col1*, creating a search range of one column.

In the FCH Utility, if *col2* is not specified, it defaults to the record length, creating a search range from *col1* to the end of the record.

#ALL Each field is searched according to its template attributes.

ref A field reference, specifying the field to be included in the search, for example: `#3`.

When the field is an array, to search all items in the array, simply specify the field reference without a subscript. To search an item in a single-dimension array, specify a subscript in parentheses to identify the occurrence that you want to find; for example: `#5(3)`. If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array; for example: `#7(2,3)`.

Multiple field references must either be enclosed in parentheses, or separated with commas but without any intervening spaces. For details, see “Limiting the search to specified fields” on page 111.

In SNGL or TABL display format, if you specify field references or do not specify a field reference or column range, when searching a numeric field, the search string is interpreted as a numeric value, and a numeric comparison is performed. When searching a character field, a string comparison is performed. That is, the search is performed based on the File Manager template attributes. For details, see “Searching numeric fields” on page 85.

ref_1 The first field reference of a range of fields. It cannot be subscripted. If the *ref_1* field reference value is less than the lowest displayed field reference value, the lowest displayed field reference value is used.

ref_2 The last field reference of a range of fields. It cannot be subscripted. If the

ref_2 field reference value is greater than the highest displayed field reference value, the highest displayed field reference value is used.

The *ref_1* and *ref_2* field reference values must be separated by a hyphen (-). Spaces are permitted between the hyphen and the field reference values.

If *ref_1* is a higher value than *ref_2*, the search process reverses the operands.

X Same as EX.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "Find/Change Utility panel" on page 574
- "Catalog Services Data Set List panel" on page 466
- "Display VTOC panel" on page 523
- "AFP Print Browse panel" on page 442
- "Memory Browse panel" on page 604
- "Field Selection/Edit panel" on page 565
- "Member Selection panel" on page 600
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Finding specific data" on page 84
- "Finding and changing data in multiple PDS members" on page 270
- "Refining the list of selected members" on page 272
- "Specifying multiple arguments for FIND" on page 276
- "Working with a list of catalog entries" on page 314
- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Browsing AFP Print documents" on page 329
- "Browsing your user storage" on page 329
- "Specifying a data set and a member name" on page 16
- "Advanced copybook selection" on page 147
- "Working with files or members containing JCL" on page 279
- "Working with WebSphere MQ" on page 331

FINDNOT primary command

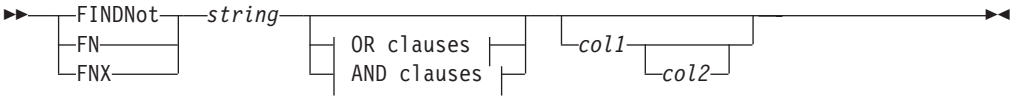
The FINDNOT command is used in the FCH Utility to return a list of members of a data set which do NOT contain a specified character string. It can only be used for a PDS or PDS/E data set.

The FINDNOT command can be abbreviated to FINDN or entered as FN. The FNX command displays an Extended Command Entry panel, in which you can enter long arguments that may not otherwise fit within the Command line (see "Handling long strings in FIND" on page 85 for more information).

Note: It is not possible to use multiple FINDNOT commands in the input stream for batch processing. Similarly, it is not possible to combine FIND and FINDNOT commands in the batch input stream.

Syntax

Syntax in the Find/Change Utility



OR clauses:

AND clauses:

string Search string. The string must be no more than 100 characters long. The string can be:

- a character string not starting or ending with a quotation mark and not containing any embedded blanks or commas. The case of the string is ignored. Uppercase and lowercase representations of the same character match. For example, Mixed string matches MIXED string.
- a character string enclosed in quotation marks. The string can contain blanks and commas. The case of the string is ignored.
- `C` followed by a character string enclosed in quotation marks (C'Frog'), or a character string enclosed in quotation marks followed by `C` ('Frog'C). The string can contain blanks and commas. The string must match exactly (including case).
- `X` followed by a hexadecimal string enclosed in quotation marks (X'C1C2').

AND and OR clauses

You can specify multiple arguments by using OR-clauses or AND-clauses. If OR-clauses are used, a member is excluded when any of the arguments are found in the member. If AND-clauses are used, a member is excluded only if all of the arguments are found in any line of the member. The maximum number of arguments that can be specified is 16.

In preparing a FINDNOT command in a batch run, you can use more than one line to specify multiple arguments. To indicate that a line is continued, the last item on the line is a blank delimited comma.

col1 and col2

You can limit the columns that are searched by the FINDNOT command by entering a pair of column numbers indicating the first and last columns

to be searched. The string is found if it is completely contained within the designated columns. For example, `FINDNOT xxx 1 20` excludes a member if it contains any records with the string "xxx" within columns 1-20.

If columns are not specified, the columns to be searched default to the columns defined by the BOUNDS line. If a single column is specified, the character string must start in the specified column. If the second column specified is larger than the record size, the record size is used.

Availability

- "Find/Change Utility panel" on page 574

Related tasks and examples

- "Finding and changing data in multiple PDS members" on page 270

FMAP primary command

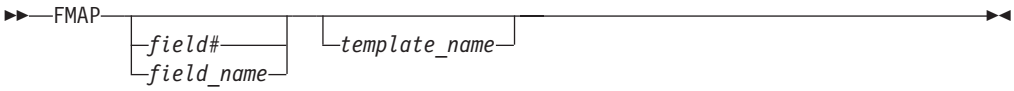
The FMAP command can be used to edit or view a field in a record using another template. If the field being mapped is shorter than the template mapping a warning message is issued when data is changed and any data which is beyond the end of the mapped field is ignored. The data type of the field being mapped is ignored.

The record containing the field to be mapped must be at the top of the screen or the cursor may be positioned on the record when the FMAP command is issued. The record must be a selected record.

If the field identification parameter is not specified and the cursor is positioned on a field that field is used. Otherwise a Field List Selection panel is displayed.

If the template name is no specified a panel is displayed where this information can be supplied.

Syntax



field# The reference number of the field to be mapped.

field_name
The name of the field to be mapped. The name may be qualified.

template_name
The name of the template to be used to map the field. A partially or fully qualified data set name together with the member name can be specified.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

FORMAT primary command

The FORMAT primary command allows you to change the current display format without having to overtype the FORMAT field. If no format type is supplied, the default depends on the current display format. If you are in SNGL format, the default is TABL format. If you are in TABL format, the default is SNGL format. In other display formats, there is no default and a error message is issued when no parameter is supplied.

The abbreviated forms of the FORMAT primary command (FS, FT, FC, FH, FL) are also available as prefix commands (see "Editor panel" on page 534).

Syntax

Sngl, FS
Change format to SNGL format. Only available if a template has been used.

Tabl, FT
Change format to TABL format. Only available if a template has been used.

Char, FC
Change format to CHAR format.

Hex, FH
Change format to HEX format.

Lhex, FL
Change format to LHEX format.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

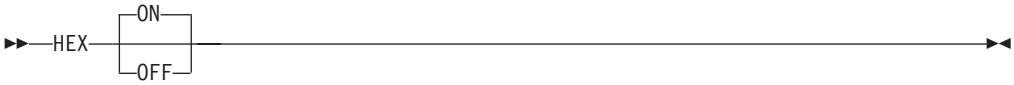
HEX primary command

In View and Edit, you can use the HEX primary command to set or reset the hexadecimal display in CHAR, HEX, SNGL or TABL display formats. This command is not available in LHEX display format.

In WebSphere MQ selection lists, you can use the HEX primary command to set or reset the hexadecimal display of the displayed data.

Syntax

Syntax



(no parameter)
Toggles the correct hexadecimal-display status. That is, it turns hexadecimal display ON if currently OFF, and turns hexadecimal display OFF if currently ON.

ON Display the hexadecimal representation of the data vertically (three lines per byte).

OFF Display the data in character format.

In View and Edit, the effect of issuing the HEX command depends upon the current display format when you issue the command, as follows:

- When the current display format is CHAR or HEX:

  HEX ON
  Changes the display format to HEX. This is equivalent to entering HEX in the **Format** field.

  HEX OFF
  Changes the display format to CHAR. This is equivalent to entering CHAR in the **Format** field.

- When the current display format is SNGL or TABL:

  HEX ON
  Changes the display of each field to show the vertical hexadecimal representation of each field in the two lines immediately below the formatted field.

  HEX OFF
  Changes the display back to standard SNGL or TABL format, without hexadecimal representations.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Selecting a display format" on page 66
- "Working with WebSphere MQ" on page 331

INFO primary command

The INFO primary command returns you from the VSAM Extent Detail or Association Information panel, to the catalog Entry Details panel for the current data type.

Syntax

Availability

- "AIX Association Information panel" on page 446
- "Non-VSAM Association Information panel" on page 607
- "VSAM Association Information panel" on page 719
- "VSAM Statistics and Extent Detail panel" on page 726

Related tasks and examples

- "Viewing association information" on page 316
- "DSI (Data Set Information)" on page 933

JOIN primary command

Joins two lines: either the line containing the cursor and the next line (which then overlays the line starting at the cursor position); or the line displayed at the top of the screen with the following line, at the current column position. See "Splitting and joining lines" on page 123.

Syntax

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Splitting and joining lines" on page 123

JUST primary command

In an editor session, the JUST command temporarily toggles numeric fields to left or right justified when in SNGL display mode.

Syntax

Note: When numeric fields are left justified, a space is allowed on the left for sign characters (+ or -). When the number is unsigned or formatted so that the sign appears to the right of the number, a blank character is used as a placeholder on

the left, so that numeric fields align neatly (but 1 character to the right of alpha fields). Fields with an alphanumeric type such as AN or ZA are always fully left aligned, even when they contain numeric characters. These field types are not affected by the JUST command and do not allow the space for the sign character. For example, in the following display, the first field is fully left aligned because it is an alphanumeric field, the second field shows the sign character, and the numbers in the last field are aligned with the numbers in the second field even though the sign character is not displayed:

```
Typ Start   Len   Data
AN     26     2   02
ZD     28     4   -654
AN     32    10   John
AN     42    10   Brown
ZD     52     4    1875
```

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

KEY primary command

You use the KEY primary command to specify a starting position when browsing, viewing, or editing a KSDS.

The KEY primary command positions to the record containing, or beginning with, the specified keyvalue. This is similar to the ability provided by the **Key** field on the editor entry panels.

Syntax

Syntax

keyvalue

In a KSDS, File Manager positions to the record containing, or beginning with, this value.

To specify blanks or special characters, enclose the value in quotes. For example:

```
KEY '02 A'
```

For undisplayable characters, you can specify the value in hexadecimal. For example:

```
KEY 'F0F240C1'X
```

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "View panel" on page 707

Related tasks and examples

- "Starting an editor session without using templates" on page 51

LEFT primary command

You can use the LEFT primary command in:
- View or Edit
- A WebSphere MQ selection list

In View or Edit, any display format apart from SNGL, the LEFT primary command scrolls to the left of your data. (This command is not valid in SNGL mode.)

The amount (number of columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax

Syntax in View or Edit

n

In View or Edit
In CHAR, HEX or LHEX display format, scrolls left *n* columns. In TABL display format, scrolls left *n* fields.

In a WebSphere MQ selection list
Scrolls left *n* columns.

CSR Scrolls left to the cursor position.

DATA

In View or Edit
In TABL display format, acts the same as **PAGE**. In other display formats, scrolls left one column less than a page of data.

In a WebSphere MQ selection list
Acts the same as **PAGE**.

HALF Scrolls left half a page of data.

MAX

In View or Edit
In CHAR, HEX or LHEX display format, scrolls to leftmost column. In TABL display format, scrolls to the leftmost field.

In a WebSphere MQ selection list
Scrolls to leftmost column.

PAGE Scrolls left one page of data.

Scroll field
Scrolls left by the amount indicated in the **Scroll field**. This is the default if no parameter is used. Entering a parameter other than MAX in the scroll field changes the scroll field default. If MAX is entered, the scroll field reverts to the previous setting after the command is issued.

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

LIBLIST primary command

The LIBLIST primary command displays the current library list and allows you to change it if necessary. If you remove a library that has been referenced, then the copybooks referring to that library are revalidated to determine whether they exist in the new library list. If they exist, the library reference is changed to the first library in which they are found. If they are not found, they are flagged as in error.

Syntax

Availability

- "Copybook Selection panel" on page 505

Related tasks and examples

- "Advanced copybook selection" on page 147

LIST primary command

The LIST primary command prints a listing of the current format definitions for the Set DBCS Options.

Syntax

Availability

- "Set DBCS Format panel" on page 651

Related tasks and examples

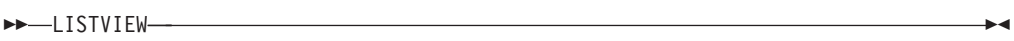
- "Printing DBCS data" on page 305

LISTVIEW primary command

On the Personal Data Set List panel the LISTVIEW primary command toggles the format of the display to include (or exclude) an extra (third) line where you can add or view the description for that data set entry.

On the Personal Data Set Lists panel, the LISTVIEW primary command toggles between the standard and extended list format. The extended list format shows the first 6 data set names for each list.

Syntax



Availability

- "Personal Data Set List panel" on page 612
- "Personal Data Set Lists panel" on page 614

Related tasks and examples

- "Working with data set lists" on page 235

LOCATE primary command

The LOCATE primary command searches the current context for an occurrence of a specified item (for example, a member, a record, a field or a line).

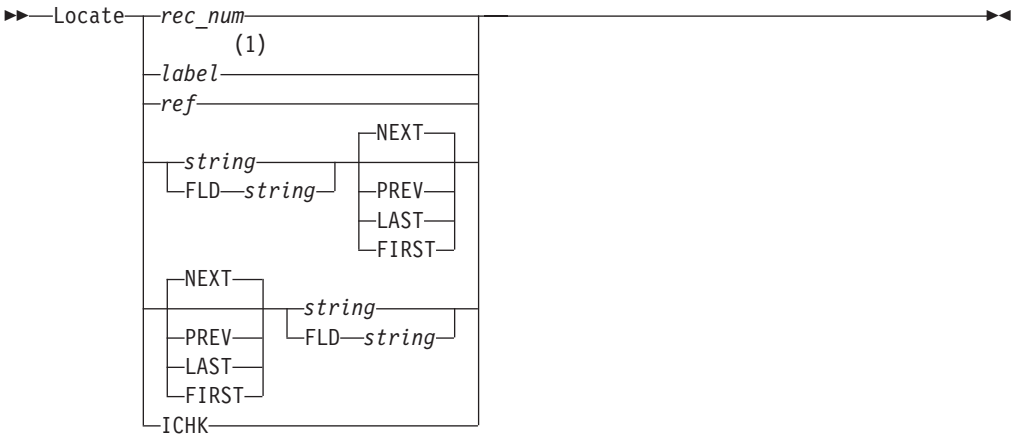
Syntax

LOCATE command in View or Edit

In an editor session, this command scrolls to the specified record or field within the displayed data set or data set member.

In an editor session, the syntax of this command is:

Syntax in View or Edit



Notes:

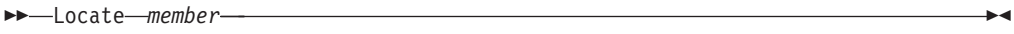
1 Edit only

LOCATE command in the Find/Change Utility

In the Find/Change Utility, this command positions a specified member at the top of the list of members to be processed.

In the Find/Change Utility, the syntax of this command is:

Syntax in the Find/Change Utility



Non-VSAM Extent Information panel

In the Non-VSAM Extent Information panel, use this command to locate specic data.

In the Non-VSAM Extent Information panel, the syntax of this command is:

Syntax in Non-VSAM Extent Information panel

LOCATE command in AFP Browse

In the AFP Browse panel, the LOCATE primary command positions a specified page number at the top of the display.

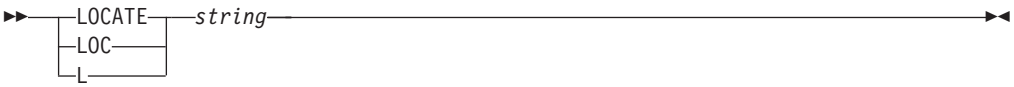
In the AFP Browse panel, the syntax of this command is:

Syntax in AFP Browse

LOCATE command in member selection panels

In member selection panels, the LOCATE command searches the current primary sort order column for a matching string to the string entered with the command. The LOCATE command positions to the top of the display the column value either equal to, or closest value less than or greater than depending on the sort order.

Syntax in Member Selection List



LOCATE command in the Copybook Selection panel

In the Copybook Selection panel, the LOCATE command searches the list of members for a matching string to the string entered with the command and positions the first matching member to the top of the display.

Syntax in the Copybook Selection panel

LOCATE command in the Field Selection/Edit panel

In the Field Selection/Edit panel, the LOCATE command searches the displayed template for either a field reference number, or the starting or full characters of a field name.

Syntax in the Field Selection/Edit panel

LOCATE command in the Load Module Information panel

In the **Load Module Information Panel**, the LOCATE command locates a specified name (if the result is sorted by name) or address (if the result is sorted by address).

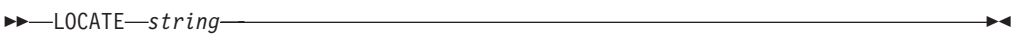
Syntax in the Load Module Information panel

LOCATE command in a WebSphere MQ selection list

When you are viewing a WebSphere MQ selection list, the LOCATE command searches the current *held* columns for a matching string to the string entered with the command.

The LOCATE command positions to the top of the display the column value which is equal to the specified string, or contains the specified string.

Syntax in in a WebSphere MQ selection list



rec_num
The number of the record that you want to locate. For example:

```
LOCATE 23
```

locates record number 23.

line_num
The line number of the data set name that you want to locate. For example:

```
LOCATE 23
```

locates the data set name on line 23.

volser (Extent panels only.) The disk volume in the list that you want to locate. For example:

```
LOCATE D$ST04
```

locates disk volume D$ST04.

(*n***)** (Extent panels only.) The *n*th disk volume in the list that you want to locate. For example:

```
LOCATE (7)
```

locates the 7th disk volume in the list.

111 (Extent panels only.) The 111-th extent of the data set. For example:

```
LOCATE 248
```

locates the 248th extent of the data set.

I (Extent panels only.) Indicates you want to search the index component of VSAM data sets. For example:

```
LOCATE FIRST I
```

locates the first volser of the data set.

page_num
The page number of the data that you want to locate. For example:

```
LOCATE 23
```

scrolls to page 23 of the print data.

label An existing user-assigned or editor-assigned label, identifying the record that you want to locate. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label. For example:

```
LOC .HERE
```

locates a record marked with an existing label (.HERE), and

```
L .ZLST
```

locates the last record in the data set.

ref A field reference, specifying the field that you want to locate, for example:

```
L #3
```

When the field is an item in an array, you must specify a subscript in parentheses to identify the occurrence that you want to locate, for example:

```
L #5(3)
```

If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array, for example:

```
L #7(2,3)
```

Available in SNGL or TABL display format only.

string The name (or part of the name) of the field that you want to locate. This string can occur anywhere in the field name. For example:

```
L SAL
```

or

```
L ARY
```

locates a field named "SALARY".

For member selection list panels, the string syntax must be consistent with the target column for the locate command. For example, for dates the string must contain valid month, day, and year values.

For the Load Module Information panel, the content of the string depends on the sequence in which the information is displayed. You can sort the result by name or by address. To sort by name, the string must be a symbol name. To sort by address, the string must be a valid hex address.

In View or Edit, the LOCATE command is available in SNGL or TABL display format only.

template_ref
A field reference, specifying the field that you want to locate in the template displayed on the Field Selection/Edit panel.

For example, to locate the cursor in the prefix area of the line containing field reference 3, issue the command:

```
L #3
```

template_fldname
The full name, or starting characters, of a field displayed on the Field Selection/Edit panel that you want to locate in the template.

For example, to locate the cursor in the prefix area of the line containing the field with the name "SALARY", issue the command:

```
L SALARY
```

or

```
L SAL
```

FLD *string*
Indicates to File Manager that the string following the FLD keyword is the name or part of the name of the field that you want to locate. This is used to resolve any ambiguity that may exist between field names and other command parameters. For example, if your field name contained the # symbol (e.g. #Items), the command `L #Items` would result in an error, as File Manager is expecting the # symbol to be followed by a numerical field reference number. Using the command `L FLD #Items` resolves this problem.

Available in SNGL or TABL display format only.

NEXT

- For non-extent panels:

  Finds the next occurrence of the field name, to the right (TABL display) or down (SNGL display) from the cursor location. Must be used in conjunction with *string* or FLD *string*, in any order. For example:

  ```
  L NEXT SAL
  ```

  or

  ```
  L SAL NEXT
  ```

  Available in SNGL or TABL display format only.
- For extent panels:

  Finds the next volser of the data set. For example:

  ```
  L NEXT
  ```
- For the Load Module Information panel:

  Finds the next name or address, depending on the order in which the information is presented. For example, if 0000FACC was recently located,

  ```
  L NEXT
  ```

  positions at the next occurrence of the address 0000FACC.

PREV

- For non-extent panels:

  Finds the previous occurrence of the field name, to the left (TABL display) or up (SNGL display) from the cursor location. Must be used in conjunction with *string* or FLD *string*, in any order. For example:

  ```
  L PREV SAL
  ```

  or

  ```
  L SAL PREV
  ```

  Available in SNGL or TABL display format only.
- For extent panels:

Finds the previous volser of the data set. For example:

```
L PREV
```

- For Load Module Information panel:

  Finds the previous name or address, depending on the order in which the information is presented. For example, if 0000FACC was recently located,

  ```
  L PREV
  ```

  positions at the previous occurrence of this address.

LAST

- For non-extent panels:

  Finds the last occurrence of the field name, regardless of the cursor location. Must be used in conjunction with *string* or FLD *string*, in any order. For example:

  ```
  L LAST SAL
  ```

  or

  ```
  L SAL LAST
  ```

  Available in SNGL or TABL display format only.
- For extent panels:

  Finds the last volser of the data set. For example:

  ```
  L LAST
  ```
- For the Load Module Information panel:

  Finds the last name or address, depending on the order in which the information is presented. For example, if 0000FACC was recently located,

  ```
  L LAST
  ```

  positions at the last occurrence of this address.

FIRST

- For non-extent panels:

  Finds the first occurrence of the field name, regardless of the cursor location. Must be used in conjunction with *string* or FLD *string*, in any order. For example:

  ```
  L FIRST SAL
  ```

  or

  ```
  L SAL FIRST
  ```

  Available in SNGL or TABL display format only.
- For extent panels:

  Finds the first volser of the data set. For example:

  ```
  L FIRST
  ```
- For the Load Module Information panel:

  Finds the first name or address, depending on the order in which the information is presented. For example, if 0000FACC was recently located,

  ```
  L FIRST
  ```

  positions at the first occurrence of this address.

ICHK Scrolls to the next record that encountered an integrity check while being saved.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "Find/Change Utility panel" on page 574
- "Catalog Services Data Set List panel" on page 466
- "Display VTOC panel" on page 523
- "AFP Print Browse panel" on page 442
- "Member Selection panel" on page 600
- "Copybook Selection panel" on page 505
- "Non-VSAM Extent Information panel" on page 610
- "VSAM Statistics and Extent Detail panel" on page 726
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Specifying a data set and a member name" on page 16
- "Advanced copybook selection" on page 147
- "Working with WebSphere MQ" on page 331

NEXT primary command

The NEXT primary command moves to the next "visible" record (see Note below) or, if you specify a number, moves forward that number of records.

Note: In this context, "visible" means any selected, not suppressed, or not excluded record, unless those types of records are being shown in the view because of edit settings or entered SHOW commands.

Note: If you specify a number of records greater than the remaining number of records, the `**** End of data ****` line is displayed.

NEXT can be abbreviated to N, NE or NEX.

Syntax

num_records
Scrolls forward the specified number of records. The default is 1 record.

MAX Scrolls to last record.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Scrolling to see data" on page 77
- "Zooming in to see all of a record" on page 71

NEXTREC primary command

When viewing segmented data in SNGL display mode, the NEXTREC primary command moves to the first segment of the next "visible" physical record (see Note below) or, if you specify a number, moves forward that number of physical records.

Note: In this context, "visible" means any selected, not suppressed, or not excluded record, unless those types of records are being shown in the view because of edit settings or entered SHOW commands.

Note: If you specify a number of records greater than the remaining number of physical records, the `**** End of data ****` line is displayed.

Syntax

num_records
Scrolls forward the specified number of physical records. The default is 1 physical record.

MAX Scrolls to first segment of the last physical record.

Availability

- "Browse panel" on page 455 (SNGL display mode only)
- "Editor panel" on page 534 (SNGL display mode only)

Related tasks and examples

- "Viewing segmented data" on page 229

NRETRIEV primary command

You use the NRETRIEV primary command to retrieve the first data set name (the one most recently used, shown at the top of the list) and associated information from the current data set list. Successive use of NRETRIEV retrieves the details for the next data set in the current data set list,

Syntax

Availability

- "Browse Entry panel" on page 459
- "Edit Entry panel" on page 541

Related tasks and examples

- "Working with data set lists" on page 235

OFFSET primary command

The OFFSET primary command is available for editor sessions that use a template. The command adds a value to the record length of the Level 01 field and to the starting position of all fields within the record type, shifting the layout left or right in relation to the records being processed. An offset value of 0 removes a previously supplied offset.

If a positive offset is applied, some fields might be mapped beyond the length of the record. Data that falls beyond the length of the record is not represented. You cannot edit a field where the data you supply would fall beyond the physical end of the record (whether the record is of fixed or variable length).

A negative offset moves the fields to the left and all fields with a resultant zero or negative start location are removed from the display. The offset is validated to ensure that required fields, such as OCCURS DEPENDING ON target fields or PL/I REFER fields, are not removed. If the first element of a dimensioned field is at a zero or negative start location, that field (all array elements) is removed from the display.

If an OFFSET primary command is applied when an offset has already been supplied in the template, the offset value overwrites the offset value provided in the template.

Syntax

value The offset value is a negative or positive integer, between -32760 and +32760, that shifts the layout left or right in relation to the records being processed. An offset value of 0 removes a previously supplied offset.

fieldname
The Level 01 field name to which the offset is applied.

The default is the currently displayed Level 01 field.

Current 01
The currently displayed Level 01 field to which the offset is applied.

ALL The offset value is applied to all the layouts described in the currently loaded template.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

- "Record Type Selection panel" on page 636
- "Copybook View and Print: Entry panel" on page 507
- "Copybook View and Print: View panel" on page 509

Related tasks and examples

- "Adjusting your view to allow for header information" on page 231
- "Specifying offset values" on page 206
- "Copybook View and Print Utility (option 3.13 or 7.2)" on page 193

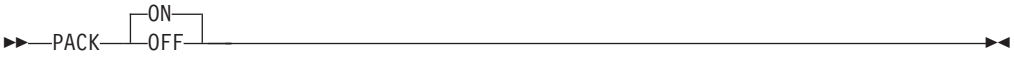
While OFFSET is not available as a standalone function, the equivalent keywords OFFSETIN and OFFSETOUT are available in the following functions:

- "DSB (Data Set Browse)" on page 868
- "DSC (Data Set Copy)" on page 875
- "DSE (Data Set Edit)" on page 900
- "DSEB (Data Set Edit Batch) — batch only" on page 907
- "DSG (Data Set Generate)" on page 922
- "DSM (Data Set Compare)" on page 934 (equivalent keywords are OFFSETOLD and OFFSETNEW).
- "DSP (Data Set Print)" on page 968
- "DSU (Data Set Update) — batch only" on page 984
- "DSV (Data Set View)" on page 998

PACK primary command

You can use the PACK primary command in the Edit panel, to convert non-packed data to the ISPF PACK format or to unpack data that is in the ISPF PACK format.

Syntax



ON Instructs File Manager to write the output file with ISPF PACK format when saving or exiting. ON is the default if no argument is supplied.

Note: If the data is already packed, and the **Recognize and interpret packed data** option is not turned on, issuing this command causes File Manager to "double-pack" the data.

OFF Instructs File Manager not to use ISPF PACK format when writing the output file. If the data was in ISPF PACK format, this unpacks the data.

Availability

- "Editor panel" on page 534

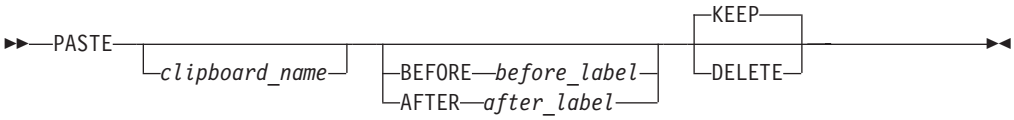
Related tasks and examples

- "Viewing and changing packed data" on page 56

PASTE primary command

The PASTE command copies lines of data from a clipboard to the current edit or view session.

Syntax



clipname
: Name of a clipboard from which data is to be copied. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

before_label
: Label identifying the line in the current edit or view session after which the data from the clipboard is to be copied. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

after_label
: Label identifying the line in the current edit or view session before which the data from the clipboard is to be copied. The label must start with a period (.) followed by one to four alphabetic characters (no numeric or special characters). Labels starting with the letter "Z" indicate an editor-assigned label.

KEEP Copied data from the clipboard remains on the clipboard.

DELETE
: Copied data from the clipboard removed from the clipboard.

Availability

- "Browse Entry panel" on page 459
- "Editor panel" on page 534

Related tasks and examples

- "Copying data to and from a clipboard" on page 121

PIC primary command

In an editor session, the PIC command toggles on and off the display of the picture clause information when in SNGL display mode. Also shows the length and scale (if non zero) for binary and packed fields, and the bit length for bit fields.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

PREFIX primary command

The PREFIX primary command sets the position and display state of the prefix area.

Syntax

LEFT Displays the prefix area on the left side.

RIGHT
Displays the prefix area on the right side.

OFF Does not display the prefix area.

ON Displays the prefix area at the position last set, with the display width last set.

n Displays the prefix area at the position last set, with a display width of *n* digits. *n* must be in the range: 6–9.

A Displays the prefix area at the position last set, with a display width of 6 digits or, if necessary, larger (up to 9 digits) in order to display the whole record number.

Availability

- "Editor panel" on page 534

PREVIOUS primary command

The PREVIOUS primary command moves to the previous "visible" record (see Note below) or, if you specify a number, moves back that number of records.

Note: In this context, "visible" means any selected, not suppressed, or not excluded record, unless those types of records are being shown in the view because of edit settings or entered SHOW commands.

Note: If you specify a number of records greater than the number of prior records, the first selected record is displayed.

Syntax

num_records
Scrolls back the specified number of records. The default is 1 record.

MAX Scrolls to first record.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Scrolling to see data" on page 77
- "Zooming in to see all of a record" on page 71

PREVREC primary command

When viewing segmented data in SNGL display mode, the PREVREC primary command moves to the first segment of the previous "visible" physical record or, if you specify a number, moves back that number of physical records.

Note: In this context, "visible" means any selected, not suppressed, or not excluded record, unless those types of records are being shown in the view because of edit settings or entered SHOW commands.

Syntax

num_records
Scrolls back the specified number of physical records. The default is 1 physical record.

MAX Scrolls to the beginning of the data set.

Availability

- "Browse panel" on page 455 (SNGL display mode only)
- "Editor panel" on page 534 (SNGL display mode only)

Related tasks and examples

- "Viewing segmented data" on page 229

PROFILE primary command

The PROFILE primary command displays the current settings in the editor session. It is similar to the ISPF PROFILE command. To remove the output lines, issue the RESET primary command or, if in an edit session, delete the lines.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Displaying your current editor options" on page 59
- "Editor options (option 0.6)" on page 45

QUIT primary command

The QUIT command is synonymous with the CANCEL command. See "CANCEL primary command" on page 744 for details.

RBALEN primary command

The RBALEN command provides a way to display or not display the RBA and record length information when viewing a VSAM file in a Browse session. The display of this information is initially controlled by the **Show RBA and Length when browsing VSAM** option in the Editor Options panel.

Syntax

ON When you are browsing a VSAM file, the leftmost side of the multiline displays contain the RBA and the length of the record being displayed.

OFF The RBA and the length of the record being displayed are not displayed, allowing more of the record data to be displayed on the screen.

Note: The RBALEN command is only available when browsing a VSAM file within a Browse session. You can specify either ON or OFF, regardless of the setting of the **Show RBA and Length when browsing VSAM** option in the Editor Options panel. You can issue the command multiple times.

Availability

- "Browse panel" on page 455
- "Editor Options panel" on page 546

Related tasks and examples

- "Displaying the RBA and record length information" on page 61

RCHANGE primary command

The RCHANGE command repeats the previous CHANGE command.

Syntax

Availability

- "Editor panel" on page 534
- "Find/Change Utility panel" on page 574

Related tasks and examples

- "Finding and replacing strings" on page 104
- "Finding and changing data in multiple PDS members" on page 270
- "Handling long strings in CHANGE" on page 107
- "Using File Manager functions in batch jobs" on page 401

RD primary command

The RD primary command prints the current record.

Syntax

RD command in Disk Browse, Disk Track Edit and VSAM Update

In these panels, the RD ("Record Dump") command prints the current record in dump format, with hexadecimal values under the record data.

Syntax - Disk Browse, Disk Track Edit and VSAM Update

RD command in View and Edit

When you are in the View and Edit panels and you are using SNGL or TABL display format, the RD command is identical to the RP command. For details, see "RP primary command" on page 811.

In other display formats, the RD ("Record Dump") command prints the current record in dump format, with hexadecimal values under the record data.

When issued from the View or Edit panel, additional parameters allow you to print a specified number of records, from the current record to the end of the data, or all displayed records.

Syntax - View and Edit

current_rec
The current record is the record in which your cursor is positioned within the data area. If your cursor is not in a record, the current record is the topmost record displayed in the data area.

n_recs Prints the next *n* displayed records, starting from the current record.

MAX Prints all remaining displayed records, from the current record to the end of the data set or member.

ALL Prints all displayed records in the data set or member.

Note: In this context, "displayed" records are those records that are currently displayable, given the settings of the Show and Shadow commands. This includes shadow lines, that is, the "substitute" records that are displayed in place of one or more records that are suppressed, excluded or not selected. For the sake of the *n* count, a shadow line is counted as one record, even if it represents a number of suppressed, excluded or not selected records.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- Disk Browse panel
- Disk Track Edit panel
- VSAM Update panel

Related tasks and examples

- "Printing a single record" on page 306
- "Disk Browse (option 5.1)" on page 371
- "Disk Track Edit (option 5.2)" on page 372
- "VSAM Update (option 5.7)" on page 380
- Chapter 8, "Tape Specific Functions," on page 337

RDF primary command

In an editor session, this command shows or hides the Redefines information and redefines fields when in SNGL display mode and shows or hides the redefines fields when in TABL display mode.

Syntax

RDF In SNGL display mode, this toggle command shows or hides the redefines definition information, all fields with a REDEFINES clause on them, and

all of their children. In TBLB display mode this command shows or hides all fields with a REDEFINES clause on them, and all of their children.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

RECLEN primary command

In multi-line formats (TABL, CHAR, HEX, and LHEX), the RECLEN primary command sets the position and display state of the RECLEN area.

Syntax

LEFT In multi-line formats, displays the RECLEN area on the left side.

RIGHT
In multi-line formats, displays the RECLEN area on the right side.

OFF Does not display the RECLEN area.

ON In multi-line formats, displays the RECLEN area at the position last set.

display-width
In multi-line formats, controls the width of the record length display area. Valid values: 3, 4, 5, 6.

A In multi-line formats, sets the width of the record length display area according to the value of the record lengths.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

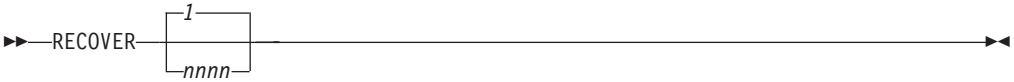
RECOVER primary command

The RECOVER primary command restores lines of data that you previously deleted during the current edit session.

Note:

1. During an Auxiliary Edit, the RECOVER command can only restore up to ten previously deleted lines of data.
2. The lines of data are restored at the point following the current top line on the panel. This may not be their original position prior to being deleted from the data set.

Syntax



nnnn Restore *nnnn* lines of data.

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Deleting records" on page 116
- "Recovering deleted records" on page 117

RECSTATS primary command

The RECSTATS primary command lists statistics for the records included in the current editor session.

Syntax

Availability

- "Editor panel" on page 534
- "Browse panel" on page 455

Related tasks and examples

- "Listing statistics for the current editor session" on page 58

REFRESH primary command

In the Catalog Services: Data Set List panel and Display VTOC panel:

- The REFRESH primary command replaces the current list with a new list taken from the catalog or VTOC. The new list incorporates any changes that you have made, and also any changes made by other users since you first displayed the list or last refreshed it.
- You can also invoke REFRESH by pressing the Process function key (F6) to display the Process pull-down menu and selecting "Refresh".

In member selection panels, the REFRESH primary command re-reads the directory and displays the current member list.

In WebSphere MQ selection lists, the REFRESH primary command re-reads and rebuilds the information required to display the selection list. After the command completes, you are returned to the top and leftmost column of the display (your current position is lost).

Syntax

Availability

- "Catalog Services Data Set List panel" on page 466
- "Display VTOC panel" on page 523
- "Member Selection panel" on page 600
- "Specifying a data set and a member name" on page 16
- "Advanced copybook selection" on page 147
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Working with a list of catalog entries" on page 314
- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Working with WebSphere MQ" on page 331

REFS primary command

The REFS primary command turns on or off the display of the Field Reference column, when in SNGL display mode.

Syntax

REFS Toggle command that shows or hides the Field Reference column when in SNGL display mode.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

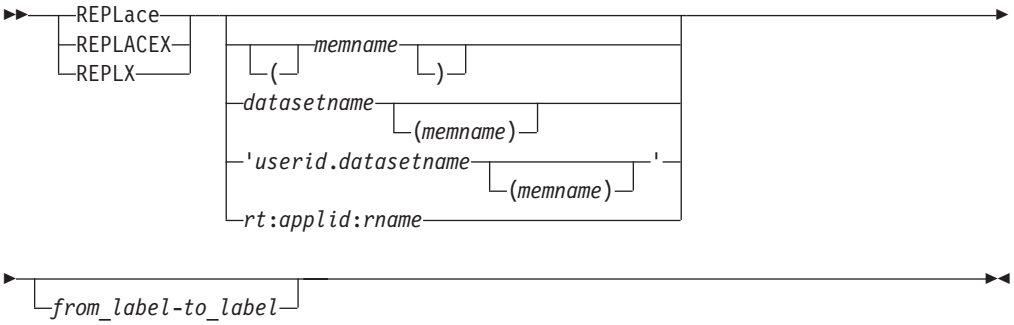
Related tasks and examples

- "Selecting a display format" on page 66

REPLACE, REPLACEX primary commands

The REPLACE and REPLACEX primary commands replace another member or data set with specified lines from the data in the current editor session.

Syntax



memname
Name of member to be replaced.

datasetname
Name of data set to be replaced. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

userid.datasetname
Name of fully-qualified data set to be replaced.

rt:*applid*:*rname*
You can specify a CICS Transient Data Queue or CICS Temporary Storage Queue in place of a data set name, where:

rt Resource type. Valid values are:
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

from_label
Label indicating first line to be copied into the member or data set.

to_label
Label indicating last line to be copied into the member or data set.

Availability

Available on the edit and view panels. MQ queues are not available as the target of this command.

Related tasks and examples

- Chapter 6, "Managing data sets," on page 235

RESET primary command

The RESET command has different functionality, depending upon where it is used.

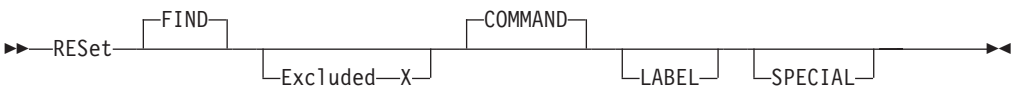
Syntax

RESET in View or Edit

In the View or Edit panels, the RESET command "cleans up" the displayed data. You use RESET to turn off the highlighting of strings found by the FIND command

and, when using a template, to restore the state of the SHOW and SHADOW settings to the defaults as set in the Editor Options panels.

Syntax



COMMAND
Resets all pending prefix commands.

EXCLUDED
"Unexcludes" (redisplays) all excluded records.

FIND Removes highlighting from strings or numeric values found by the FIND command.

LABEL
Removes all labels from labeled lines.

SPECIAL
Removes all special lines, such as the BOUNDS line from the display.

Entering a RESET command without any parameters:
- Resets everything except record labels (equivalent to entering RESET COMMAND EXCLUDED FIND SPECIAL), and
- Groups or ungroups not-selected and suppressed records and hides or displays shadow lines according to the current editor options settings.

RESET in member selection panels

In member selection panels the RESET command resets values entered in the prefix area that have not yet been processed and also in the **Prompt** field. You can specify a member name or a member pattern to limit the scope of the command to matching member names.

Syntax - member selection panels

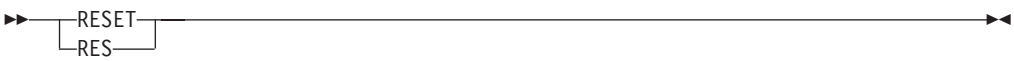
* Specifies that all members in the list are to be deselected.

member_name
Specifies that the named member is to be deselected. Can be a member name pattern, so that all members with a matching name are deselected.

RESET in Set DBCS Options

In the Set DBCS Options panel, the RESET primary command resets the format definitions to their default value (1,*,EB).

Syntax (for Set DBCS Options)



RESET in selection lists

In selection lists, the RESET primary command removes all tailorings that affect the selection list.

Syntax (for selection lists)

RESET in options panels

In the options panels, the RESET command resets the options displayed on the panel to the default options defined in the File Manager options module.

Syntax (for options panels)

Availability

- "Set Print Processing Options panel (option 0.1)" on page 658
- "Set System Processing Options panel (option 0.2)" on page 663
- "Set Tape Processing Options panel (option 0.3)" on page 665
- "Editor Options panel" on page 546
- "VSAM Edit Sharing Options panel" on page 721
- "Set Temporary Data Set Allocation Options panel" on page 667
- "Set Output Data Set Allocation Options panel" on page 655
- "Set Trace options panel" on page 669
- "Set COBOL Processing Options panel (option 0.5.2)" on page 649
- "Set HLASM Processing Options panel (option 0.5.3)" on page 652
- "Set PL/I Processing Options panel (option 0.5.4)" on page 656

Related tasks and examples

- "Setting your default processing options" on page 41

RF primary command

(Shared files only.) Refreshes a record element from the file being operated on. It has a slightly different meaning depending on the current display format in use:

- In SNGL display format, only the record currently displayed is refreshed
- In other display formats, such as CHAR, HEX, LHEX, or TABL, all records are refreshed.

Note: Any pending changes to records are discarded during the REFRESH process.

Syntax

Availability

- "Editor panel" on page 534

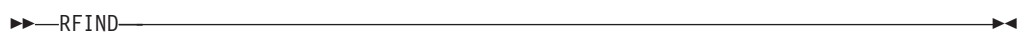
Related tasks and examples

- "Working with File Manager and shared files" on page 89

RFIND primary command

The RFIND primary command repeats the search performed by the previous FIND primary command.

Syntax



When your cursor is positioned in the body of the data, RFIND finds the next instance of *string* down from the cursor position. When your cursor is outside of the data area, RFIND finds the first instance of *string*. If you press the RFIND function key (F5), you can easily move to the each instance of *string*, as your cursor remains in the data area after the command is issued. However, if you type the RFIND command on the Command line, you need to reposition your cursor on the last instance of *string* and then press Enter, in order to find the next instance.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "Find/Change Utility panel" on page 574
- "Catalog Services Data Set List panel" on page 466
- "Display VTOC panel" on page 523
- "AFP Print Browse panel" on page 442
- "Memory Browse panel" on page 604
- "Display VTOC Data Set List panel" on page 528
- "Volume Summary/Selection panel" on page 718
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Finding specific data" on page 84
- "Finding and changing data in multiple PDS members" on page 270
- "Working with a list of catalog entries" on page 314
- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Browsing AFP Print documents" on page 329
- "Browsing your user storage" on page 329
- "Working with WebSphere MQ" on page 331

RIGHT primary command

You can use the RIGHT primary command in:
- View or Edit
- A WebSphere MQ selection list

In View or Edit, any display format apart from SNGL, the RIGHT primary command scrolls to the right of your data. (This command is not valid in SNGL mode.)

The amount (number of columns) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax

n

In View or Edit
In CHAR, HEX or LHEX display format, scrolls right *n* columns. In TABL display format, scrolls right *n* fields.

In a WebSphere MQ selection list
Scrolls right *n* columns.

CSR Scrolls right to the cursor position.

DATA

In View or Edit
In TABL display format, acts the same as **PAGE**. In other display formats, scrolls right one column less than a page of data.

In a WebSphere MQ selection list
Acts the same as **PAGE**.

HALF Scrolls right half a page of data.

MAX

In View or Edit
In CHAR, HEX or LHEX display format, scrolls to rightmost column. In TABL display format, scrolls to the rightmost field.

In a WebSphere MQ selection list
Scrolls to rightmost column.

PAGE Scrolls right one page of data.

Scroll field
Scrolls right by the amount indicated in the **Scroll field**. This is the default

if no parameter is used. Entering a parameter other than MAX in the scroll field changes the scroll field default. If MAX is entered, the scroll field reverts to the previous setting after the command is issued.

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

RP primary command

The RP ("Record Print") command prints the current record. The format of the output from the RP command depends on the display format when you enter the RP command.

Syntax

RP command in Disk Browse, Disk Track Edit or VSAM Update

Syntax - Disk Browse, Disk Track Edit or VSAM Update

RP command in View or Edit

When issued from the View or Edit panel, additional parameters allow you to print a specified number of records, from the current record to the end of the data, or all displayed records.

Syntax

current_rec
The current record is the record in which your cursor is positioned within the data area. If your cursor is not in a record, the current record is the topmost record displayed in the data area.

n_recs Prints the next *n* displayed records, starting from the current record.

MAX Prints all remaining displayed records, from the current record to the end of the data set or member.

ALL Prints all displayed records in the data set or member.

Note: In this context, "displayed" records are those records that are currently displayable, given the settings of the Show and Shadow commands. This includes shadow lines, that is, the "substitute" records that are displayed in place of one or more records that are suppressed, excluded or not selected. For the sake of the *n* count, a shadow line is counted as one record, even if it represents a number of suppressed, excluded or not selected records.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- Disk Browse panel
- Disk Track Edit panel
- VSAM Update panel

Related tasks and examples

- "Printing a single record" on page 306
- "Disk Browse (option 5.1)" on page 371
- "Disk Track Edit (option 5.2)" on page 372
- "VSAM Update (option 5.7)" on page 380
- Chapter 8, "Tape Specific Functions," on page 337

RUNTEMP primary command

The RUNTEMP primary command runs the current function with the template changes you have made but does not save the changes. This command is only available when running a function that uses a template.

Syntax

Availability

- "Field Selection/Edit panel" on page 565

Related tasks and examples

- "Creating a copybook template with the Template Workbench" on page 150

SAVE primary command

The SAVE primary command saves any changes made to a data set without ending the Edit session.

Note: This command is not supported when editing a large file using an auxiliary data set. With auxiliary editing, you must save the data by ending the edit session with the END, EXIT or FILE command, and then, if required, re-edit the data set.

Syntax

Availability

- "Editor panel" on page 534
- "Dynamic Template panel" on page 529
- "Record Identification Criteria panel" on page 625
- "Record Selection Criteria panel" on page 631
- "Field Selection/Edit panel" on page 565

Related tasks and examples

- "Saving changes without ending the Edit session" on page 60

SAVEAS primary command (templates)

The SAVEAS primary command (for templates) displays the Save panel, where you can save a template as another name.

Syntax

Availability

- "Field Selection/Edit panel" on page 565

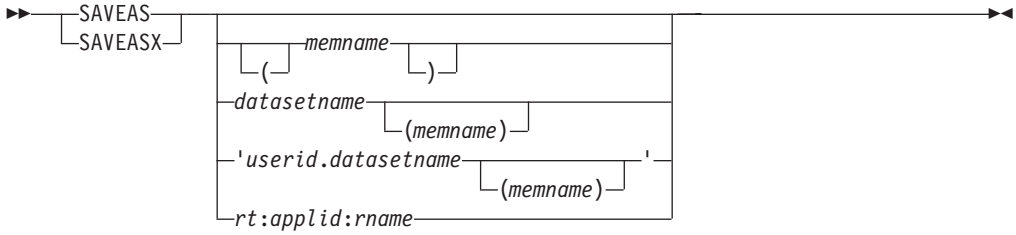
Related tasks and examples

- "Editing a template" on page 156

SAVEAS, SAVEASX primary commands (data)

The SAVEAS and SAVEASX primary commands (for data) save the currently exposed data in the current editor session to another member or data set.

Syntax



memname
Name of member to be created.

datasetname
Name of data set to be created. The default High Level Qualifier (HLQ) is used. Typically, the default is the TSO PREFIX for your user ID.

userid.datasetname
Name of fully-qualified data set to be created.

rt:*applid*:*rname*
You can specify a CICS Transient Data Queue or CICS Temporary Storage Queue in place of a data set name, where:

rt Resource type. Valid values are:
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

Availability

Available on the edit and view panels. MQ queues are not available as the target for this command.

Related tasks and examples

• Chapter 6, “Managing data sets,” on page 235

SELECT primary command

The SELECT primary command is used to select members or fields, depending upon where it is used. It can be entered on the following panels:

- Member selection
- Field Selection/Edit

Syntax

SELECT in member selection panels

The SELECT primary command is used to select members listed in member selection panels. It performs the equivalent processing of the S prefix command against member names matching the supplied pattern or member name.

SELECT results in one of the following:
- A single member is selected and returned to the calling routine. (Subsequent selections are ignored.)
- Toggling of the selected value in the **Prompt** field.
- Selects a member to be processed by the function that invoked the member list.

You can clear the selection of members with the RESET command.

Syntax for member selection panels:

Syntax - Member Selection panel

* Specifies that all members in the list are to be selected.

member_name
Specifies that the named member is to be selected. Can be a member name pattern, to select all members with a matching name.

HOLD
Default. Specifies that after selecting one or all members you wish to pause before processing commences. Whilst in HOLD mode, you can scroll UP and DOWN to view selected members and select or deselect members to refine your list. Pressing Enter causes processing to commence. You can use the LOCATE command whilst in pause mode and processing does not commence until you press Enter again.

Note: For functions where the SELECT command toggles selection, this parameter is ignored and the behaviour is always NOHOLD.

NOHOLD
Specifies that processing is to be run immediately.

Note: For functions where the SELECT command toggles selection, this parameter is ignored and the behaviour is always NOHOLD.

Examples

SELECT MBR
Selects the single member, MBR.

SELECT *
Selects all the listed members.

SELECT a*
Selects all the listed members starting with the letter "A".

SELECT *z
Selects all the listed members ending with the letter "Z".

SELECT in Field Selection/Edit

In the Field Selection/Edit panel, this command selects or deselects all of the listed fields.

Syntax - Field Selection/Edit panel

ALL Denotes all fields.

ON Select all fields.

OFF Deselect all fields.

Availability

- "Member Selection panel" on page 600
- "Field Selection/Edit panel" on page 565

Related tasks and examples

- "Specifying a data set and a member name" on page 16
- "Advanced copybook selection" on page 147

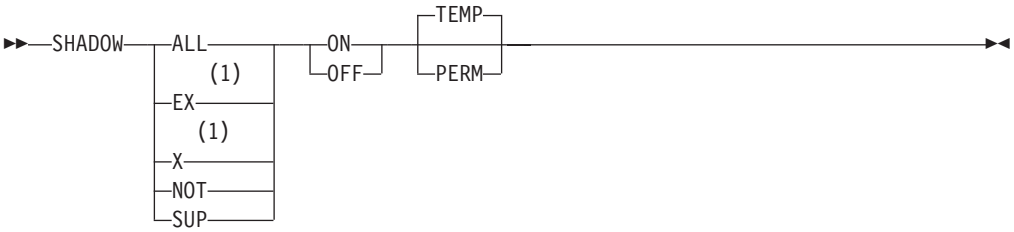
SHADOW primary command

The SHADOW primary command hides or shows shadow lines. A shadow line is a line that represent a group of records that are not-selected, suppressed, or have been excluded by the EXCLUDE command.

For a description of not-selected and suppressed records, see "Filtering record display using templates" on page 224.

For information about using the EXCLUDE command, see "EXCLUDE/XX primary command" on page 762.

Syntax



Notes:

1 Edit only

ALL Show or hide all types of shadow lines.

EX Show or hide shadow lines for "excluded" records.

NOT Where supported, show or hide not-selected records shadow lines.

OFF Hide the specified types of shadow lines.

ON Show the specified types of shadow lines.

PERM The shadow setting is saved in your user profile, and applies to this and future File Manager sessions until changed. Issuing the command changes the equivalent **Show shadow lines...** settings on the Editor Options (option 0.8) panel.

SUP Where supported, show or hide suppressed records shadow lines.

TEMP Shadow setting applies to the current Edit session only. This is the default.

X Same as EX.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Filtering record display using templates" on page 224
- "Displaying "suppressed" records" on page 224
- "Displaying "not-selected" records" on page 226

SHOW primary command

Use the SHOW command to expose or group *not-selected* and *suppressed* records: for a definition of these terms, see "Filtering record display using templates" on page 224. Records that have been grouped using the SHOW command can be represented by shadow lines. You can display or hide the shadow lines with the SHADOW command.

The SHOW command lets you see records that, by default, would not be displayed. Using the SHOW command, you can review records that contain erroneous or corrupt data.

When the display format is TABL, only fields from one record type can be displayed at any one time. Because of this, even with SHOW SUP ON, it is not possible to format records that have a different record type than the one currently being displayed. Instead, each record belonging to another record type is represented by a line (similar to a shadow line) that contains the name of the record type for that record.

If you show not-selected records in SNGL or TABL display format, then File Manager attempts to interpret and format data according to the field definitions for the current record type.

Syntax

ALL Show or group both not-selected and suppressed records.

NOT Show or group not-selected records.

SUP Show or group suppressed records.

ON Show the specified types of records in addition to records currently shown.

OFF Group the specified types of records.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Filtering record display using templates" on page 224
- "Displaying "suppressed" records" on page 224
- "Displaying "not-selected" records" on page 226

SHOWCOB primary command

The SHOWCOB command displays details showing which COBOL compiler is currently being used.

Syntax

Availability

Available on all panels

Related tasks and examples

- "Checking which COBOL compiler you're using" on page 13

SLOC primary command

In an editor session, the SLOC command toggles on and off the display of the Start location of each field when in SNGL display mode.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

SORT primary command

The SORT command changes the order of the data you are viewing.

In an editor session, by default, File Manager sorts data in ascending order. For VTOC data sets and member lists, the default sequence depends on the sort criteria you specify.

Syntax

SORT command in View or Edit

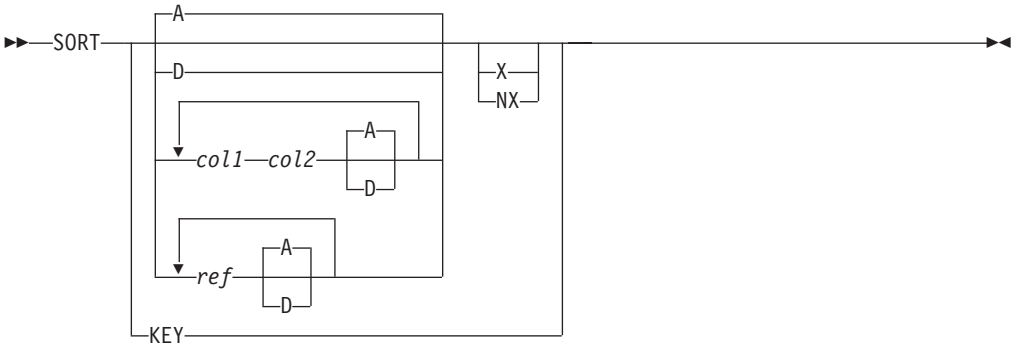
In View and Edit, File Manager sorts the data using a hierarchy of keys starting with the first column or field parameter as the primary key, and subsequent columns or fields as less significant keys. The hierarchy is specified left to right.

If you do not specify any parameters for the SORT primary command, File Manager sorts the data record by record, according to each record's hexadecimal representation.

The sort order is based on EBCDIC (hex) values in the specified columns or fields, therefore, lowercase characters are sorted before uppercase characters.

When you use the SORT primary command, you can optionally specify that only excluded, or only not-excluded, records are sorted.

Syntax: View and Edit



A Sort data in ascending sequence. This is the default.

D Sort data in descending sequence.

col1 (CHAR, HEX, or LHEX display formats only.) The first column to be included in the column range to be used as a sort key. Must be greater than or equal to 1, and less than or equal to the maximum record length. You can specify up to five pairs of column range values.

col2 (CHAR, HEX, or LHEX display formats only.) The last column to be included in the column range to be used as a sort key. Must be greater than *col1* and less than or equal to the maximum record length. If *col1* is specified, then *col2* must also be specified.

ref (TABL and SNGL display format only.) A field reference, specifying the field to be used as the sort key, for example: `#3`. To sort a TABL display format, you *must* provide at least one field reference.

When the field is an item in an array, you must specify a subscript in parentheses to identify the occurrence that you want to use, for example: `#5(3)`. If your field is part of a multi-dimensional array, you must specify a subscript for each dimension in the array, for example `#7(2,3)`.

You can specify up to 5 field references. Multiple field references must be separated by a blank, for example: `#3 #5(3)`. The first reference specifies the primary (most significant) key, the second reference specifies the secondary (2nd most significant) key, and so on. Sorting proceeds from the most significant key to the least significant key.

SORT does not support fields with varying length or varying starting positions. If one of these fields is specified, an error message is displayed.

NX Not-excluded records only. Sorts records as if any excluded records did not exist. If you subsequently display excluded records, they may be are interspersed with the sorted non-excluded records.

X Excluded records only. Sorts excluded records as if any not-excluded records did not exist. If you subsequently display non-excluded records, they may be are interspersed with the sorted excluded records.

KEY Sort a KSDS, VRDS, or RRDS VSAM data set into the KEY sequence of the file. Typically you use SORT KEY after performing other sort operations on the data. The SORT KEY function “restores” the records into the original key sequence of the file.

This option only sorts non-excluded records. So `SORT KEY` is the equivalent of `SORT KEY NX` (even though you cannot specify the NX).

Note: When you sort a RRDS file using SORT KEY, File Manager sorts the RRDS into VSAM SLOT sequence. The SLOT number is a unique number that the VSAM access method allocates for each record. If the records of the RRDS file have been re-ordered by moving records via a move (M, M*n*, or MM) prefix command or SORT primary command, SORT KEY re-orders the records as they were when the file was opened. However, when you *add* records using insert (I or I*n*), copy (C, C*n*, or CC), or repeat (R, R*n*, RR, or RR*n*) prefix commands, the new records do not have a SLOT number assigned as the file is still in memory. In this case, SORT KEY re-orders the new records at the bottom of the file.

SORT command in selection list panels

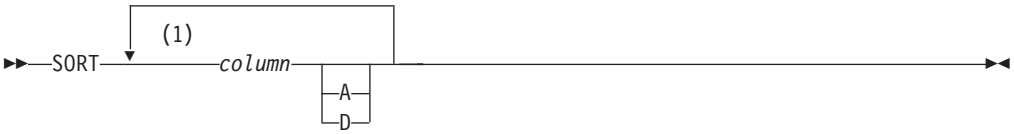
You can use the SORT command to sort specified column names in a panel showing a selection list.

When more than one column is to be sorted, the sequence in which you specify the columns has no impact on the sequence in which the columns are sorted. The columns are always sorted from left to right.

If you do not specify a sort order (A or D) for a column, the default sort order for that column is used. You can find the default sort order for a column in the Column Settings panel. To display this panel, position the cursor on the filter value for the column and press Enter.

After the SORT command has been entered, the list display is scrolled to the top with the order reflecting the changes after applying the sort to the specified columns.

Syntax: selection list panels



Notes:

1 You can specify up to 5 columns to be sorted.

column Column to be sorted.

A Sort column in ascending order.

D Sort column in descending order

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534
- "Display VTOC Data Set List panel" on page 528
- "Volume Summary/Selection panel" on page 718
- "Member Selection panel" on page 600

Related tasks and examples

- "Sorting the data" on page 82
- "Displaying a Volume Table of Contents (VTOC)" on page 325
- "Specifying a data set and a member name" on page 16
- "Advanced copybook selection" on page 147

SPLT primary command

Splits a line in two: either the line containing the cursor, at the cursor position, or the top line at the current column position.

Syntax

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Splitting and joining lines" on page 123

SPLTJOIN primary command

Splits or joins lines, depending on the position of the cursor. If the cursor position in a line or the current column position is after the last non-blank character, a join operation is performed. Otherwise a split operation is performed. SPLTJOIN can be abbreviated to SJ.

Syntax

Availability

- "Editor panel" on page 534 (only while in CHAR, HEX, or LHEX display formats)

Related tasks and examples

- "Splitting and joining lines" on page 123

STR primary command

The STR primary command is available:

In an editor session
The STR command toggles on and off the display of the template structure information (field/element level) for each field when in SNGL display mode.

In template edit
The STR command toggles the displaying of the structure (level

information) in the Field Name area of the display. The current setting is remembered for future sessions. This command is not available when processing a dynamic template.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

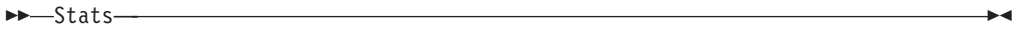
Related tasks and examples

- "Selecting a display format" on page 66

STATS primary command

The STATS primary command displays the VSAM Extent Detail panel.

Syntax



Availability

- "AIX Entry Detail panel" on page 447
- "VSAM Entry Detail panel" on page 722

Related tasks and examples

- "Statistics for VSAM data sets" on page 317

SV primary command

In SNGL display format, the SV primary command saves the changes to the current record without ending the edit session. (Shared files only.)

Syntax

Availability

- "Editor panel" on page 534

Related tasks and examples

- "Working with File Manager and shared files" on page 89

TAILOR primary command

The TAILOR command provides a list of columns and their attributes for the current selection list. You can change the attributes in order to customize or limit the display. Changeable attributes include:

The column attributes you can change are:

Order A numerical value specifying the relative order (left to right) in which the column is displayed.

If you specify a value of 0 for **Order**, the column is suppressed from being displayed.

Sort Specifies how the data in the column is to be sorted:

A Sort data in ascending order.

D Sort data in descending order.

N Do not sort the data.

Hold Specifies if the column is to be made non-scrollable:

N

Y

Width The width of the column.

Filter Restricts the data displayed in a selection list to data that matches the specified filter string.

You can specify wildcard characters in the filter.

Filtering is performed using a generic trailing match. That is, if the start of the data in the column being filtered matches the filter string, it is considered a match. For example, a filter of `PEA` matches PEA and PEAR, but not APPEAR.

Matching of data to filters is not case-sensitive. For example, a filter of `PEA` matches PEA, PEAR, Pea and pear.

You can specify any of these operators as the first character of the filter:

> Greater than.

< Less than.

= Equal to.

! Not equal to.

When a column contains a filter parameter, the title is shown in yellow.

Note: You cannot use the TAILOR primary command with the selection list displayed for the TAILOR operation.

Syntax

Availability

- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Working with WebSphere MQ" on page 331

TEDIT primary command

The TEDIT command displays different template editing panels, depending upon the type of template currently in use. When you have finished editing the template, you return to viewing the data set.

Syntax

If you are using a dynamic template or a copybook template with only one record type that has been identified, the TEDIT command displays the Field Selection/Edit panel, in which you can select fields and specify record selection criteria.

If you are using a copybook template with more than one record type that has been identified, the TEDIT command displays the Record Type Selection panel, in which you can select or edit records by type.

If you are not currently using a template, the TEDIT command acts like the TVIEW command, and shows the Template Workbench main panel, where you can create, select, and optionally edit a template before returning to viewing the data set with the template.

You can abbreviate the command TEDIT to TE.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Editing a template" on page 156

TOP primary command

The TOP primary command scrolls to the first page of data. You can abbreviate the command TOP to T.

Syntax

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729

- "WebSphere MQ Queue List panel" on page 736

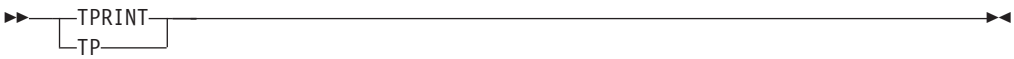
Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

TPRINT primary command

The TPRINT primary command sends the current template to the output destination specified in your Print Processing Options.

Syntax



Availability

- "Field Selection/Edit panel" on page 565

Related tasks and examples

TVIEW primary command

The TVIEW command shows the Template Workbench, where you can create, select, and optionally edit a template before returning to browsing the data set with the template.

Use the TVIEW command:

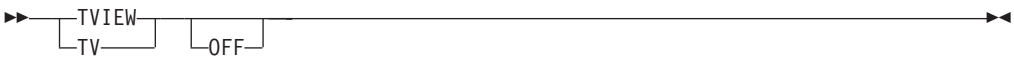
- If you are already using a template, but want to switch to using a different template.
- If you are not currently using a template, and want to create or select a template.

You can abbreviate the command TVIEW to TV.

The OFF parameter disables the current view and shows all records.

Note: Issuing the TV command followed by a F3 (Exit) or F12 (Cancel) is synonymous to a TV OFF command. You must issue the RC or RT command to resume edit with a template from the workbench.

Syntax



Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Editing a template" on page 156

TYPE primary command

The TYPE primary command turns on or off the display of the Type and Len (field length) columns, when in SNGL display mode.

Syntax

TYPE Toggle command that shows or hides the Type and Len columns when in SNGL display mode.

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Selecting a display format" on page 66

UP primary command

The UP primary command scrolls up (backward) through your data.

The amount (number of lines) scrolled is determined by either an optional parameter or, if no parameter is entered, by the amount indicated in the **Scroll** field.

Syntax

nnnn Scroll up *nnnn* lines.

Csr Scroll up to the cursor position.

Data Scroll up one line less than a page of data.

Half Scroll up half a page of data.

Max Scroll to top of file. This has the same effect as the TOP command.

Page Scroll up one page of data.

Scroll field
Scroll up by the amount indicated in the **Scroll field**. This is the default if no parameter is used.

Availability

Available on all panels when the panel information cannot be seen within one screen. In particular, it is used in the following panels, to allow you to view data:

- "Browse panel" on page 455
- "Editor panel" on page 534
- "WebSphere MQ Managers panel" on page 729
- "WebSphere MQ Queue List panel" on page 736

Related tasks and examples

- "Scrolling to see data" on page 77
- "Working with WebSphere MQ" on page 331

UPDATE primary command

The UPDATE primary command compiles the specified copybook and either updates an existing template or creates a new template.

Syntax

Availability

- "Copybook Selection panel" on page 505

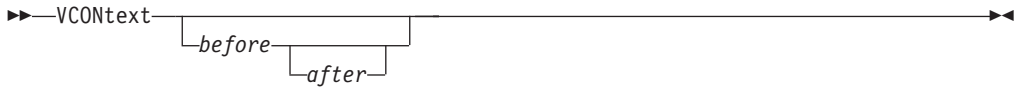
Related tasks and examples

- "Advanced copybook selection" on page 147

VCONTEXT primary command

The VCONTEXT primary command is used in the FCH Utility panel to change the output display of the search results. When the LIST=LONG option is specified for a FIND, FINDNOT or CHANGE command, the records containing the target string are displayed in the output listing. The VCONTEXT command, which can be entered as VCON, allows the "found" record to be displayed in context by allowing the user to specify the number of records to be displayed before and after the "found" record.

Syntax



before Indicates the number of records to be displayed BEFORE the record containing the found string.

after Indicates the number of records to be displayed AFTER the record containing the found string. If *after* is not specified, it defaults to the same value as *before*.

Note: If neither value is specified, both are reset to zero.

Availability

- "Find/Change Utility panel" on page 574

Related tasks and examples

- "Specifying context in the output listing" on page 278

VER primary command

The VER command displays the File Manager release and PTF level, and indicates whether or not File Manager is AF-authorized.

If the VER command already exists in the ISPF command table at your site, you can use the LVL synonym to perform the same task.

Syntax

Availability

Available on all panels

Related tasks and examples

- "Checking your File Manager version" on page 11
- "VER (Display Service Level)" on page 1085

VIEW primary command

The VIEW primary command displays currently suppressed records of the record type indicated by the shadow line selected by the cursor. Records of other record types are suppressed from display. To use the VIEW command, type `VIEW` on the Command line, position the cursor on the shadow line of the suppressed records you want to view, then press Enter.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Changing display of the record type" on page 225

VOLUME primary command

The VOLUME primary command displays volume information for a VSAM or non-VSAM data set The information is displayed in a pop-up Volume Information panel showing a list of allocated volumes for the selected data set.

Syntax

Availability

- "VSAM Entry Detail panel" on page 722
- "Non-VSAM Entry Detail panel" on page 609

Related tasks and examples

- "Displaying volume information" on page 317

XD primary command

The XD command converts a hexadecimal value to its equivalent decimal value.

Syntax

Availability

Available on all panels.

Results

Related tasks and examples

"Using hexadecimal values in File Manager" on page 37

ZOOM primary command

In CHAR, HEX and LHEX display formats, the ZOOM primary command limits the display to just one record, showing you all of the data in the record without having to scroll left or right.

In SNGL and TABL display formats, zooming in shows all fields in the record presented in SNGL display format, rather than only those fields selected by the template.

The ZOOM command zooms in on the record selected by the cursor, or the top record displayed if the cursor is not positioned on a record.

To zoom back out to the display format that was current prior to zooming in, enter the ZOOM command again.

Syntax

Availability

- "Browse panel" on page 455
- "Editor panel" on page 534

Related tasks and examples

- "Zooming in to see all of a record" on page 71
- "Changing the length of a record" on page 125

Chapter 16. Functions

This chapter contains information about the File Manager functions that you can use to perform File Manager tasks. It also contains information about the File Manager-supplied REXX external functions that can be used in REXX procedures and template criteria expressions run within File Manager.

The File Manager functions can be invoked from batch jobs using JCL, within TSO as commands, and from REXX and CLIST procedures. The REXX external functions are only made available to a REXX environment running under the control of a File Manager function.

How to use this reference chapter

This chapter lists the File Manager functions in alphabetical order, followed by the REXX external functions in alphabetical order.

The following subsections are supplied, where relevant, for each function:

Purpose
A short summary of what you can do with the function.

Usage notes
A more detailed explanation of the function.

Options
A more detailed explanation of the function.

Return Codes
Lists the return codes that are specific to the function.

Related functions
Other functions that are related in some way (for example, if the function is for tapes, the list of other tape functions).

Syntax
A syntax diagram, followed by a parameter list. The parameter list describes the parameter, and shows (where appropriate) its maximum value and default value.

If you specify parameters that are not relevant to the function, File Manager ignores them. If you specify the same parameter more than once, File Manager uses the first value that you specify.

Batch example
A listing of a sample batch job.

How to read the syntax diagrams

Function syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

  The ►►— symbol indicates the beginning of a statement.

  The —► symbol indicates that a statement is continued on the next line.

  The ►— symbol indicates that a statement is continued from the previous line.

The ⟶◂ symbol indicates the end of a statement.

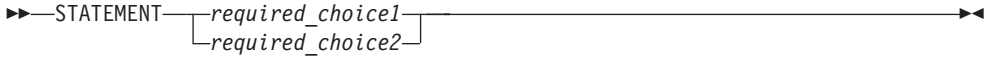
- Required items appear on the horizontal line (the main path).

- Optional items appear below the main path.

- If a default value is provided for an item (that is, if no parameter is specified, the default is used), the default item appears above the main path.

- If you can choose from two or more items, they appear vertically, in a stack.

  If you *must* choose one of the items, one item of the stack appears on the main path.



  If choosing one of the items is optional, the entire stack appears below the main path.

  If choosing an item is optional, but when no selection is made, the default is used, the default appears above the line and the remaining optional choices below the line.

- An arrow returning to the left above the main line indicates an item that can be repeated.

- Keywords (for example, INPUT) appear in uppercase letters. They must be spelled exactly as shown.
- Variables (for example, *nlrecs*) appear in lowercase letters. They represent user-supplied names or values.

Related topics

"Using File Manager functions in batch jobs" on page 401

General tips about performance when you use File Manager functions

The following tips are provided as a guide to improving performance when using File Manager functions:

- **Reading sequential input files with DFSORT**
  - For the best overall File Manager product performance, it is strongly recommended, for all File Manager installations, that DFSORT be made available to File Manager so that it can use DFSORT I/O technology. For details, see the *File Manager Customization Guide*.

    Note: It is not necessary for you to have a DFSORT license in order for File Manager to use DFSORT technology.
  - DFSORT technology can be used by File Manager in interactive operations such as certain File Manager edit and browse scenarios. It can also be used by File Manager for both interactive and batch executions of the File Manager utilities, Data Set Copy (DSC) and Data Set Print (DSP).
  - For the DSC and DSP utilities, File Manager attempts to use DFSORT technology when reading:
    - Sequential (DSORG=PS) data sets.
    - A VSAM file, provided there are FASTPROC (DFSORT) commands in the user PROC. If there is no user PROC, or the PROC contains no FASTPROC statements, File Manager uses its own VSAM I/O.
  - For DSC, when File Manager chooses to use DFSORT technology to read an input file, it may also use DFSORT to write the primary output data set.
  - Development measurements show that File Manager using DFSORT technology can reduce sequential (DSORG=PS) I/O (EXCPs) by two orders of magnitude. File Manager using DFSORT technology can also significantly reduce CPU utilization, compared to File Manager without access to DFSORT technology. Both I/O and CPU performance gains are greatest in the simplest situations where little or no special File Manager processing is required. The CPU gains can also be significant when all, or a large portion, of any record reformatting or other work can be done by DFSORT using FASTPROC (DFSORT) commands.
  - File Manager can use DFSORT technology even if it needs to process individual records, such as when using a PROC or a template or copybook. However, some things can prevent File Manager from using DFSORT I/O technology on a particular sequential input file:
    - An IO exit is being used on the input data set.
    - The input data set:
      - Is a concatenated data set with unlike attributes.
      - Is an LRECL=X data set.
      - Is a HFS file.
      - Contains records of undefined length (RECFM=U).
      - Is being processed with a template that contains segmented records.

    File Manager makes the determination about DFSORT use based on the above conditions and other factors which may affect performance. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for copy processing. Such invocation may fail since DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

  - It is possible to directly control DFSORT processing using DFSORT commands in a File Manager PROC. For details, see the documentation on FASTPROC in "Coding DFSORT procedure statements" on page 409. It can be worth using this feature when very high performance is required. DFSORT can perform operations with high efficiency before File Manager processes the data records, or in some cases without File Manager processing any data records at all. The main "cost" to you, the user, is learning and using another syntax for DFSORT commands.

- **Use the FASTREXX subset of REXX**

  If a user PROC is required, it is strongly recommended that, rather than the full REXX language, the FASTREXX subset of REXX is used. File Manager provides a set of external REXX functions designed to allow many common tasks to be performed within the FASTREXX subset. See Chapter 13, "Enhancing File Manager processing," on page 405 for a discussion of FASTREXX, and "External REXX functions" on page 1094.

- **Template and copybook processing**

  The utilities DSC, DSEB, DSG, DSM, DSP, DSU, and PBK allow template or copybook processing. When using these functions, consider the following performance-related points:

  - File Manager can be most efficient if a template is used instead of a copybook.
  - If a copybook must be used, specify the language of the copybook. (LANG=COBOL or LANG=PLI, instead of the default LANG=AUTO) for best performance. If the language of the copybook is HLASM, specify LANG=HLASM.
  - Template expressions should be written, if at all possible, using expressions suitable for File Manager internal processing. For more details, see "Improving performance using internal expression processing" on page 213.

Also, see "Performance tips" for the following functions:
- "DSC (Data Set Copy)" on page 875
- "DSE (Data Set Edit)" on page 900
- "DSM (Data Set Compare)" on page 934
- "DSP (Data Set Print)" on page 968
- "DSU (Data Set Update) — batch only" on page 984
- "FCH (Find/Change)" on page 1015

File Manager functions

This section describes the File Manager functions that you can use in batch jobs, in TSO, and in REXX and CLIST procedures. For more information about using these functions, see Chapter 12, "Introduction to programming with File Manager functions," on page 397.

AUD (Print Audit Trail Report)

Purpose

Print a formatted or unformatted audit trail report.

Usage Notes

You must specify the name of the audit trail data set from which you want to produce a report. If the audit trail is from FM/IMS or FM/DB2, you must also indicate that.

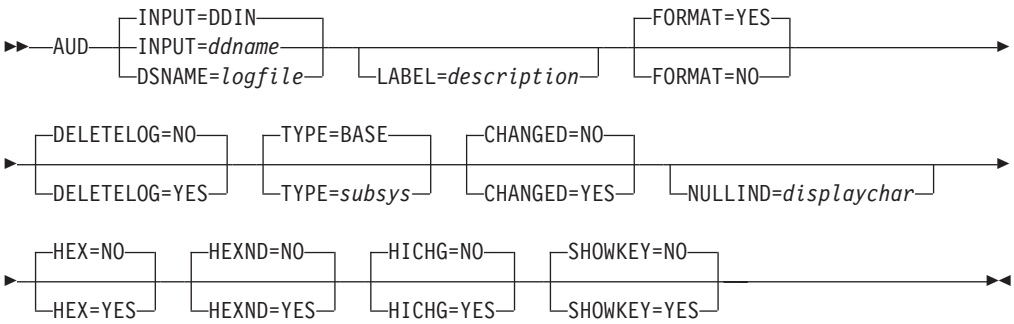
Options

You can choose between printing an unformatted report or, if the audited function used a template, a report formatted according to the template you used. You can also specify whether or not you want the audit trail data set to be deleted after printing. You can provide a description to help to identify the audit trail report. For FM/DB2, you can specify any non-standard null display character used.

Related functions

None.

Syntax



INPUT=*ddname*

Defines a reference to a DD or TSO ALLOC statement for the input data set or HFS file. The default is DDIN.

DSNAME=*logfile*

Specifies the name of the audit trail log data set.

LABEL=*Description*

Optional identification for the audit trail report. Must be within quotes if it contains blanks.

FORMAT

Determines the formatting of the audit trail report.

YES Default. Report is formatted according to the template used in the audited Edit session.

NO Report is not formatted.

DELETELOG

Determines whether or not the audit trail data set is deleted after printing. DELETELOG is not allowed for log data sets allocated using a DD statement.

NO Default. The audit trail data set is not deleted.

YES The audit trail data set is deleted after the report is printed.

TYPE Specifies the subsystem used for the audited Edit session. Can be one of:

BASE Default.

IMS

DB2

CICS

CHANGED

NO Default. All fields are reported.

YES Only fields that are changed are reported.

NULLIND=*displaychar*
Specifies the null indicator character to be used in an audit trail report for DB2 data.

HEX=YES
Produces an UPDOWN hexadecimal display below the standard field display.

HEXND=YES
Produces an UPDOWN hexadecimal display below the standard field display only for fields that contain non displayable characters.

HICHG=YES
Highlights the changed fields. An asterisk is placed to the left of the "before" data to indicate the field has been changed.

SHOWKEY=YES
Displays key fields even when CHANGED=YES has been selected. A "K" is printed to the left of key field names. For a KSDS data set, a key field is any elementary field that intersects, or is contained in, the key area. Note: does not apply to DB2 audit reports.

```
//AUDIT job (acct),'name'
//* Print formatted audit trail report
//*
//FMNBATCH EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
$$FILEM AUD ,
$$FILEM DSNAME=FMUSER.FMNLOG.D020816.T070549,
$$FILEM LABEL='my description - BASE',
$$FILEM FORMAT=YES,
$$FILEM DELETELOG=NO
/*
//
```

```
//AUDIT job (acct),'name'
//* Print formatted audit trail report
//*
//FMNBATCH EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
$$FILEM AUD ,
$$FILEM TYPE=IMS,
$$FILEM DSNAME=FMUSER.IMSAUDIT.D021012.T085732,
$$FILEM LABEL='my description - IMS',
$$FILEM FORMAT=YES,
$$FILEM DELETELOG=NO
/*
//
```

```
//AUDIT job (acct),'name'
//* Print formatted audit trail report
//*
//FMNBATCH EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
```

```
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
$$FILEM AUD ,
$$FILEM TYPE=DB2,
$$FILEM DSNAME=FMUSER.FMN2AUD.DBT6.D021012.T141043,
$$FILEM LABEL='my description - DB2',
$$FILEM NULLIND=_,
$$FILEM DELETELOG=NO
/*
//
```

BSF (Backward Space File)

Purpose

Move the tape backward one or more tape files.

Usage notes

A tape file consists of blocks of data delimited by a tape mark. For this function, a tape label set is also considered a file. The tape stops after the specified number of tape marks has been read or the load point has been reached. The tape is positioned immediately before a tape mark or at the load point.

Specify a DDNAME for the tape.

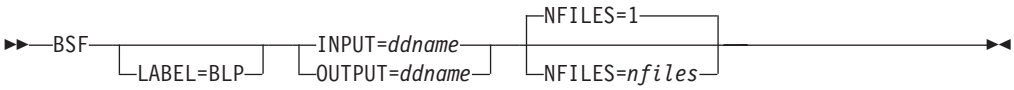
Options

None.

Related functions

BSR Move the tape backward one or more records
FSF Move the tape forward one or more files
FSR Move the tape forward one or more records
REW Rewind a tape to its load point
RUN Rewind a tape and unload it

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *File Manager Customization Guide*.

nfiles Number of tape files to be spaced backward. The maximum is 99 999; the default is 1.

BSR (Backward Space Record)

Purpose

Move the tape backward one or more physical records (blocks).

Usage notes

A physical tape record is either a block of data bytes or a tape mark. A

tape mark is treated as a special record. For each tape mark crossed, File Manager displays an information message.

Specify a DDNAME for the tape.

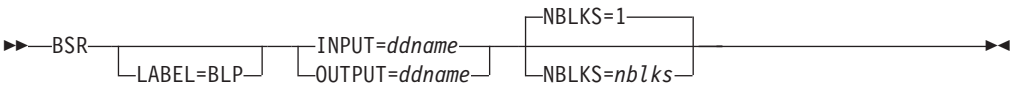
Options

None.

Related functions

BSF Move the tape backward one or more files
FSR Move the tape forward one or more records
FSF Move the tape forward one or more files
REW Rewind a tape to its load point
RUN Rewind a tape and unload it

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nblks Number of tape blocks to be spaced backward. The maximum is 99 999 999; the default is 1. A tape mark is considered a block.

BT (Create Tape File)

Purpose

Write test data to a tape file.

Usage notes

Use this function to format a tape or create test data.

Specify a DDNAME for the tape. Also specify the record size and the number of records to be written.

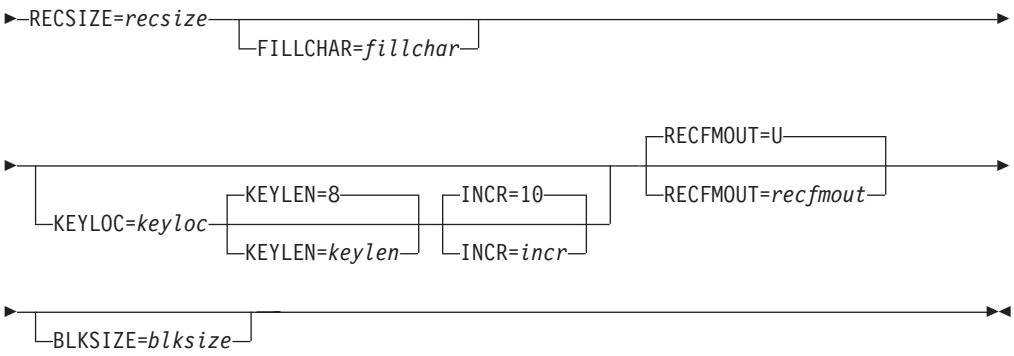
Options

You can specify a fill character or fill pattern, a sequence field, a record format, and a block size.

Related functions

DSG Write test data to a data set
INT Initialize a tape

Syntax



blksize
If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape is processed by other utilities or standard access methods, you must also consider the operating system limits.

ddname
Refers to a DD or TSO ALLOC statement. To create a tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

fillchar
To fill each byte of each record with data, specify one of the following:

char A character, such as 0 to be written in each byte

X'*cc***'** A binary character, such as X'04' to be written in each byte

AN To write alphanumeric characters (A to Z and 0 to 9)

BIN To write binary characters (X'00' to X'FF')

RAND
To write random binary characters (X'00' to X'FF')

If you specify AN or BIN, characters are written in a "ripple" pattern. For example, if you specify AN, the first 3 records look like this:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789AB
CDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABC
```

The default is a blank.

incr Increment value for the sequence field. The default is 10.

keylen
Length of the sequence field, from 1 to 9. The default is 8.

keyloc Position (starting with 1) of the sequence field within the output records. If omitted, there is no sequence field.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

nlrecs Number of logical records to be written. The maximum is 99 999 999.

recfmout

Record format for the output. Each value is a combination of the following letters:

B Blocked

D Variable-length ISO/ANSI tape records

F Fixed length

S Spanned format

U Undefined length

V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize

Length of each logical record. The maximum is 9 999 999.

```
//BT  JOB (acct),'name'     Create SL Tape
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//*  EXAMPLE BT JOB, SL processing
//*  67 RECORDS ARE WRITTEN TO A TAPE FILE,
//*  V RECFM.  THE OUTPUT FILE CONTAINS
//*  ALPHANUMERIC CHARACTERS.  THERE IS A
//*  SEQUENCE FIELD AT POSITION 1.
//*
//STP0001     EXEC FMBAT
//BT1      DD DISP=(NEW,KEEP),UNIT=CART,
//            VOL=(,RETAIN,,,SER=FMO001),
//            LABEL=(1,SL),
//            DSN=FMNUSER.BT05.A1V
//SYSIN    DD *
$$FILEM BT  OUTPUT=BT1,NLRECS=67,RECSIZE=1024,
$$FILEM     KEYLOC=1,RECFMOUT=V,FILLCHAR=AN
$$FILEM EOJ
/*
//*
//*  12 RECORDS ARE WRITTEN TO A TAPE FILE,
//*  FB RECFM.  THE OUTPUT FILE CONTAINS '*'
//*  CHARACTERS.  THERE IS A SEQUENCE FIELD
//*  AT POSITION 10.
//*
//STP0002     EXEC FMBAT
//BT2      DD DISP=(NEW,KEEP),UNIT=CART,
//            VOL=(,RETAIN,,,SER=FMO001),
//            LABEL=(2,SL),
//            DSN=FMNUSER.BT05.A2FB
//SYSIN    DD *
$$FILEM BT  OUTPUT=BT2,NLRECS=12,RECSIZE=2048,
$$FILEM     KEYLOC=10,INCR=100,
$$FILEM     RECFMOUT=FB,FILLCHAR='*',
$$FILEM     BLKSIZE=20480
$$FILEM EOJ
/*
```

BTB (Batch Template Build)

Purpose

To produce multiple templates from a PDS of copybooks in batch.

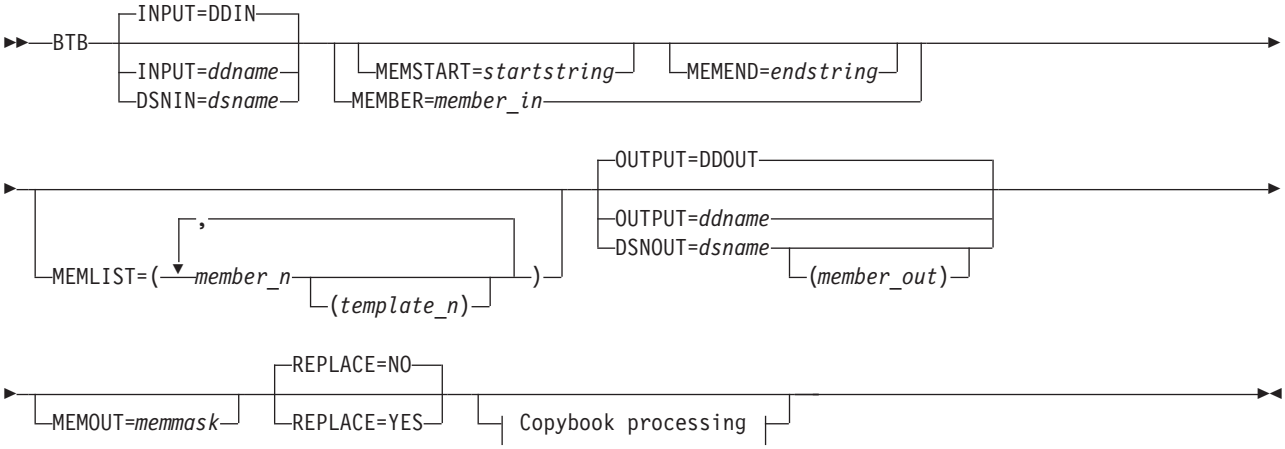
Usage notes

- You can select the copybooks from which templates are to be produced by member name selection criteria.

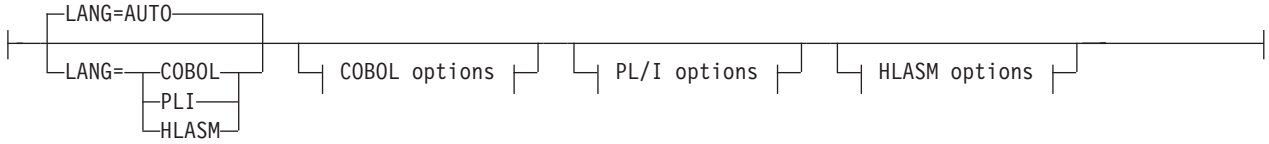
Related functions

BTU Batch template update

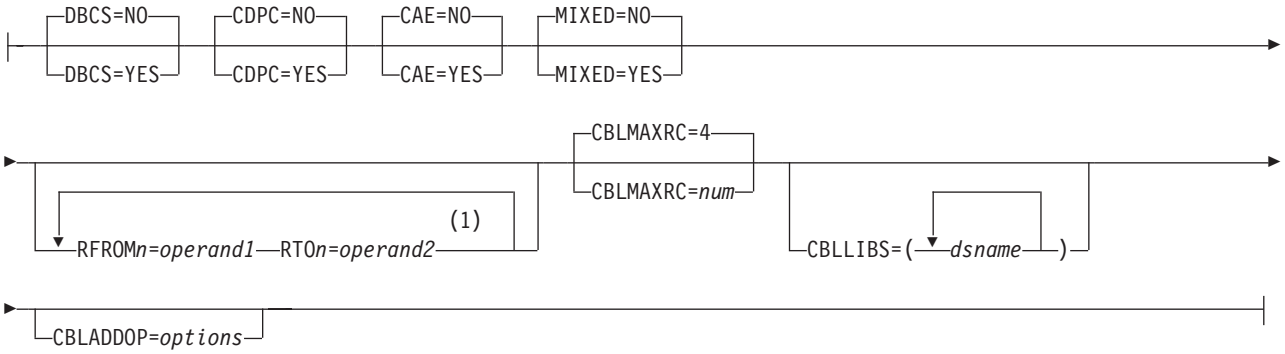
Syntax



Copybook processing:

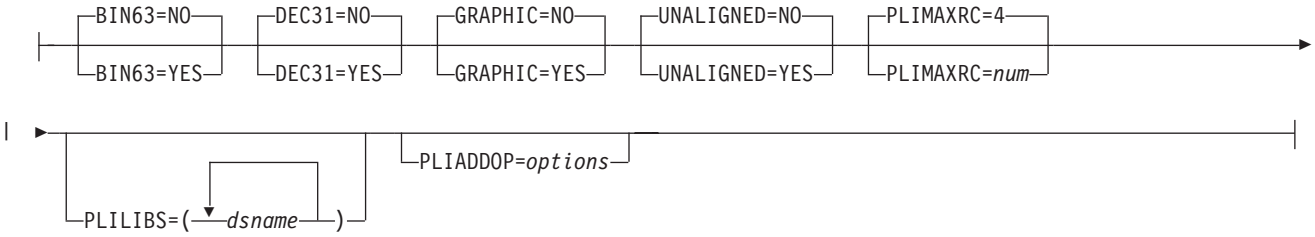


COBOL options:

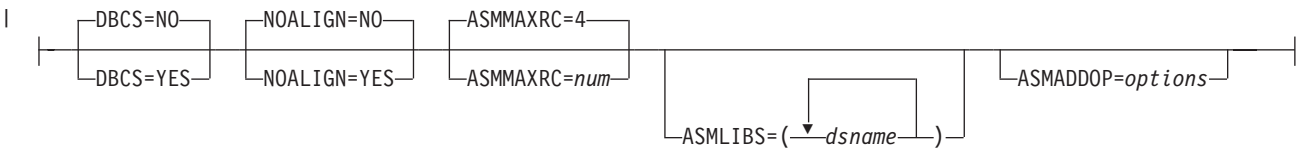


Function reference: BTB

PL/I options:



HLASM options:



Notes:

1 *n*= 1 to 5.

INPUT=*ddname*
This points to the input DD name which can have one or more associated data sets that must be valid copybook data sets. If Panvalet or Carma data sets are referred to, then they must all be Panvalet or CARMA data sets. You cannot mix. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets.

DSNIN=*dsname*
Data set name where copybooks reside.

MEMBER=*member_in*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member_in is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the

PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member-in* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member_in* parameter of the MEMBER keyword.

MEMLIST
Allows you to specify a list of member names with optional associated output template names. If you do not specify the associated template name, File Manager uses the copybook name or the name as identified by the MEMOUT mask, *memmask*.

member_n
The name of the member to be processed. Generic name masks are allowed.

template_n
The name of the template after it has been copied to the output data set. If unspecified, the output template is not renamed.

OUTPUT=ddname
Identifies the DD card which points to the template data set in which the templates are stored or replaced. It must refer to a PDS, PDSE, or sequential data set. Concatenated data sets are not supported.

DSNOUT=*dsname*
Defines the data set name where templates are created. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows:

(*member-out***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=*memmask*
Where a number of input members have been specified, you can specify a member name pattern for the output templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single

character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates in the output partitioned data set are not replaced.

YES Like-named templates in the output partitioned data set are replaced.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

```
//DDIN  DD DSN=FMNUSER.COBOL2,DISP=SHR
//      DD DSN=FMNUSER.COBOL,DISP=SHR
//SYSIN DD *
$$FILEM BTB INPUT=DDIN,
$$FILEM MEMLIST=(DITTST1,
$$FILEM          DITTST3,
$$FILEM          DITTST7,
$$FILEM          FMNTST1),
$$FILEM LANG=COBOL,
$$FILEM COMPMAXRC=08,
$$FILEM RFROM1===:==,
$$FILEM RTO1=====,
$$FILEM REPLACE=YES,
$$FILEM DSNOUT=FMNUSER.TEMPLATE.NEW
```

Will produce the following output (if there is more than one input data set):

```
Report 1 Data set list

Data set name                                    Lib
-------------------------------------------------
FMNUSER.COBOL2                                     1
FMNUSER.COBOL                                      2
```

```
Build report

          Template Build Report
Copybook   Lib Template    Status
DITTST1      1 DITTST1     Replaced
DITTST3      1 DITTST3     Replaced
DITTST7      1 DITTST7     Compile error
FMNTST1      2 FMNTST1     Created

4 members read : Template : 1 Created  2 Replaced  1 Errors
```

The Lib column is displayed only when more than one input data set is found.

BTU (Batch Template Update)

Purpose

Use this utility to:

- Update one or more templates.
- Create new templates based upon existing templates.

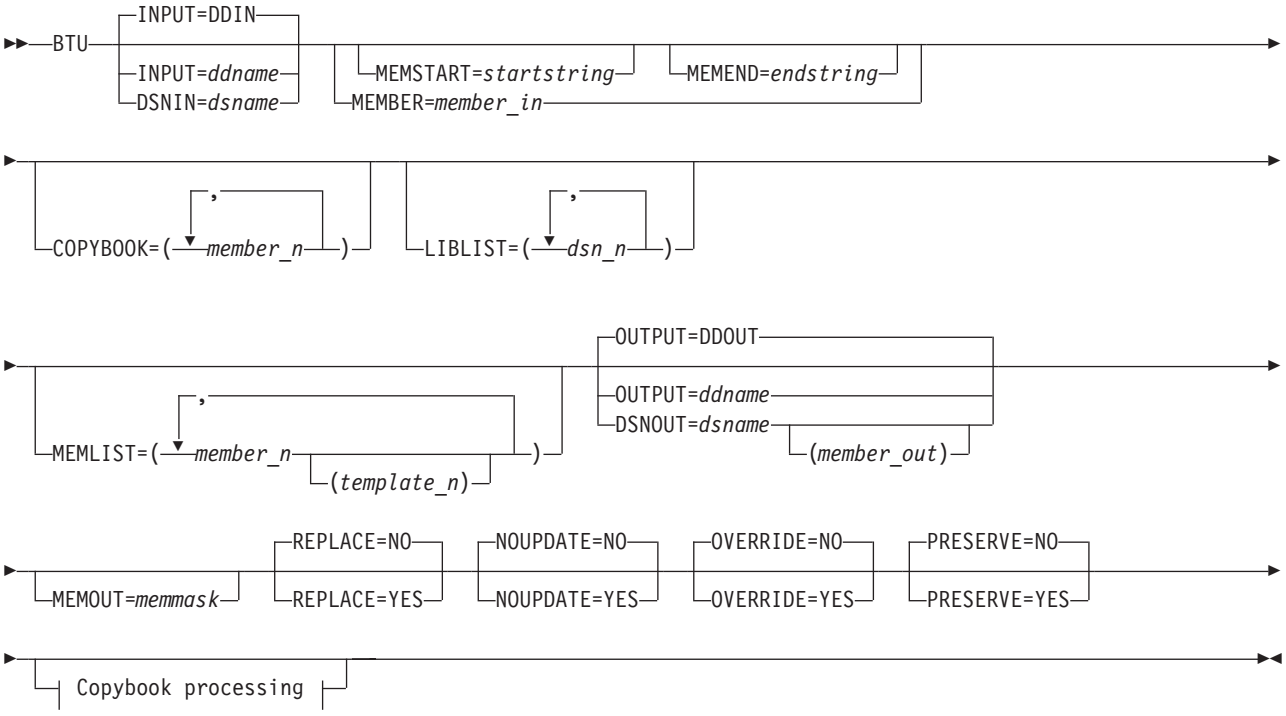
Usage notes

- You can filter the templates selected for processing by providing copybook member names or member masks so that only templates referencing such copybooks are updated.

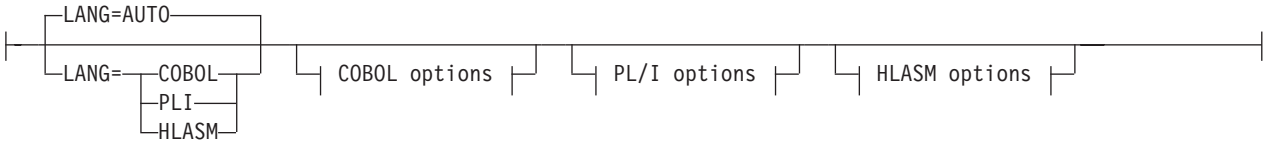
Related functions

BTB Batch template build

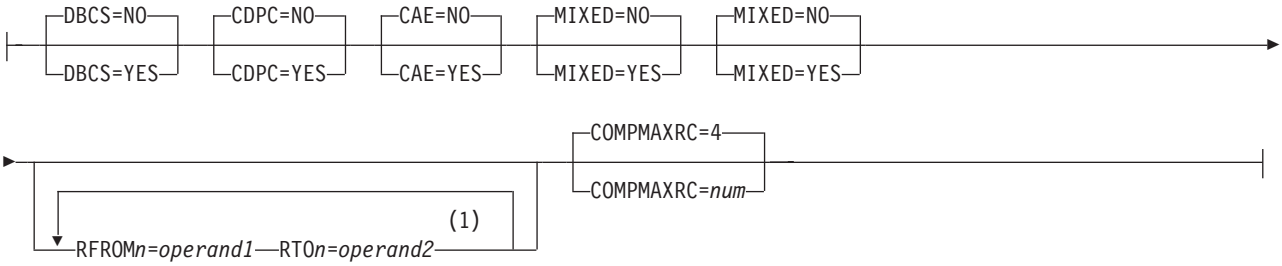
Syntax



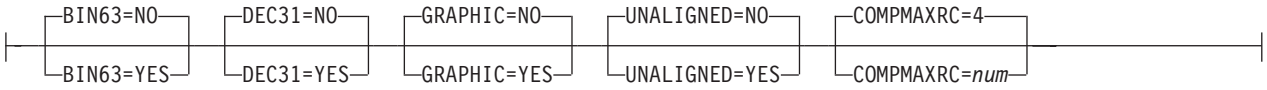
Copybook processing:



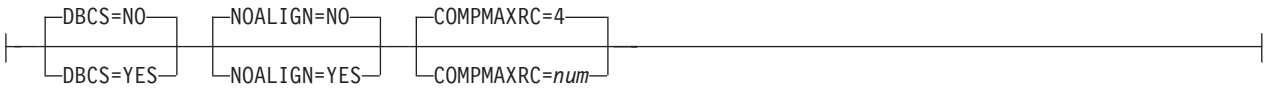
COBOL options:



PL/I options:



HLASM options:



Notes:

1 *n*= 1 to 5.

INPUT=*ddname*
This points to the input DD name which can have one or more associated data sets that must be valid template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets. You can specify concatenated data sets. If an output data set is not provided, the template is updated in the data set from which it originated.

DSNIN=*dsname*
Data set name where templates reside.

MEMBER=*member_in*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the PDS whose name is four characters in length are processed.

member_in is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member-in* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member_in* parameter of the MEMBER keyword.

COPYBOOK
Allows you to specify a list of member names or patterns used to filter so that only templates referencing those copybooks, or copybooks that match the patterns, are selected for processing.

member_n
The name of the member to be processed. Generic name masks are not allowed.

LIBLIST
Allows you to specify a list of up to twelve data set names used to replace the copybook library lists in the selected templates.

dsn_n The name of the data set name to be processed. Generic name masks are not allowed.

MEMLIST
Allows you to specify a list of member names with optional associated output template names. If you do not specify the associated template name, File Manager uses the copybook name or the name as identified by the MEMOUT mask, *memmask*.

member_n
The name of the member to be processed. Generic name masks are allowed.

template_n
The name of the template after it has been copied to the output data set. If unspecified, the output template is not renamed.

OUTPUT=ddname
Identifies the DD card which points to the template data set in which the templates are stored or replaced. It must refer to a PDS, PDSE, or sequential data set. Concatenated data sets are not supported.

DSNOUT=*dsname*
Defines the data set name where templates are created. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows:

(*member-out***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=*memmask*
Where a number of input members have been specified, you can specify a member name pattern for the output templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates in the output partitioned data set are not replaced.

YES Like-named templates in the output partitioned data set are replaced.

NOUPDATE
Specifies whether or not File Manager writes back updates to the data set.

NO Updates are written back to the data set.

YES Updates are not written back to the data set.

OVERRIDE
Specifies whether or not File Manager overrides any compile options found in the template with the compiler options found specified in the parameter list.

NO Compile options found in the template are not overwritten with the compiler options found specified in the parameter list.

YES Compile options found in the template are overwritten with the compiler options found specified in the parameter list.

PRESERVE
Specifies whether or not File Manager uses the current version of the copybook.

NO File Manager searches for the first version of the copybook.

YES File Manager uses the current version of the cookbook, provided the copybook still exists in the library it was previously found in, and the library is in the list the update process is using.

File Manager searches for the first version of the copybook in the order the libraries are listed if one of these conditions applies:

- NO has been specified for this option.
- The copybook no longer exists in the library it was previously found in.
- The library is not in the list the update process is using.

Compiler options
If you specify OVERRIDE=YES, then the specified compile options shown here override the options found in the template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

COMPMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

COMPMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

COMPMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

```
//DDIN   DD DSN=FMNUSER.TEMPLAT1,DISP=SHR
//       DD DSN=FMNUSER.TEMPLAT2,DISP=SHR
//SYSIN DD *
$$FILEM BTU INPUT=DDIN,
$$FILEM MEMLIST=(COPYTST,
$$FILEM          COPY01,
$$FILEM          COPY0102)
```

Report 1 Data set list (only produced for multiple input data sets).

```
Data set name                                    Lib
-------------------------------------------------
FMNUSER.TEMPLAT1                                   1
FMNUSER.TEMPLAT2                                   2
```

Update report

```
          Template Update Report

Template  Lib New name              Status
-------------------------------------------------------------------------------
COPYTST     1                       Updated
COPY01      1                       Not replaced
COPY0102    2                       Updated

3 members read 2 Updated 0 Not changed 1 Not replaced  0 Errors
```

Table 13. Batch update status and action

Status	Explanation	Action
Updated	The template was successfully updated.	None
Not Replaced	The template exists in the output data set and the replace option is NO.	Specify replace and rerun if required.
Replaced	The template exists in the output data set and has been successfully updated with replace option YES.	None.
Compile Error	Unable to compile the copybooks associated with the template.	Rerun using option 7.4 in foreground for the failing template and look at the compile listing produced.
Corrupt Template	The internal format of the template is corrupted. This could have occurred because the template has been modified outside of File Manager.	This is an internal error. If the template has not been modified then keep a copy of it and contact you IBM representative.
SYSLIB not found *syslib*	The syslib referenced in the template could not be found.	Rerun the update in foreground using option 7.1 which will list the SYSLIBs which you can them modify. Alternatively provide LIBLIST=(dsn1,dsn2...) parameter to identify the current location for the copybooks.
SYSLIB invalid attrs *syslib*	The syslib referenced in the template has invalid attributes for the language type for this template.	Change the syslib reference using option 7.1 or LIBLIST parameter.
Copybook not found name	The copybook name could not be located in the current libraries.	Provide the data set containing the referenced copybook using either 7.1 or the LIBLIST parameter.
Storage exhausted	File Manager ran out of storage during processing	Increase the region size.
No copybooks in libraries	The library list provided either from the template or override has no copybooks.	Rerun the update in foreground using option 7.1 which will list the SYSLIBs which you can them modify. Alternatively provide LIBLIST=(dsn1,dsn2...) parameter to identify the current location for the copybooks.
Not a valid template	The type of template is not valid for update processing.	This is an internal error that should never occur—contact your IBM support center.

Table 13. Batch update status and action (continued)

Status	Explanation	Action
Duplicate name	The output template name has already been referenced by another template during update.	Correct parameters so that there are no duplicate names being saved.
Not found	The template member referenced could be not be found on the input data set.	Correct the input parameters to point to the right data set or member name.
Save error	The updated template could not be saved	Normally a space problem—check the output data set (or input data set if an output data set was not provided) and increase the size.
Update check successful	Running with `NOUPDATE=YES` and update would have been done.	None.
Updating of criteria failed	The update was not performed because fields that were previously referenced in criteria could not be found in the current versions of the copybooks.	Rerun the update in foreground using option 7.1 and the U command. This should allow you to correct the expressions that have been invalidated as a result of your copybook changes.

CLM (Compare Load Module)

Purpose

Use the CLM function to:

- Perform a load module comparison. Load module and CSECT information from both the "Old" and "New" versions of the module is extracted and compared. By specifying compare criteria, you can see differences between specific properties of the load modules such as load module size, link date, CSECT name, and compilers used.
- Produce a comparison report, showing information such as where insertions, deletions, or changes have occurred in the "New" load module. The content and structure of the report reflect the comparison options used.

Usage notes

Specify the way in which the comparison is performed, using:

- The compare level
- The compare criteria

Specify the type of output produced and the way in which the output is displayed, using:

- The listing type
- The listing options

You can use Data Set Compare (DSM) instead of CLM to compare load modules, by specifying SYNCH=LMOD.

Return codes

The default return codes from the CLM function have the following modified meanings:

0 The function completed successfully and the compare sets match.

1 The function completed successfully and the compare sets do not match.

2 One of the compare sets was empty, so no comparison was performed.

4 Both of the compare sets were empty, so no comparison was performed.

16 Data set or member in use.

16 Data set or member open error.

16 Data set or member not found.

16 Other input or output error occurred.

16 Member name required and not specified.

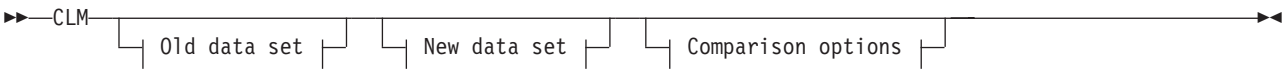
16 Insufficient storage available.

16 CLM abended.

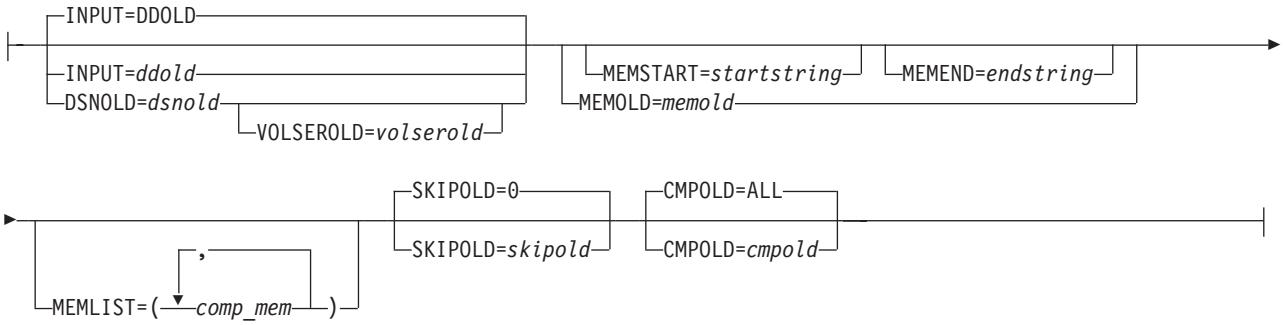
16 Other serious error that stops processing occurred.

16 A severe error occurred, causing File Manager to terminate.

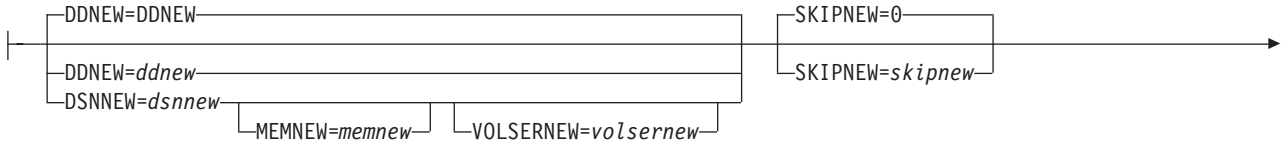
Syntax



Old data set:

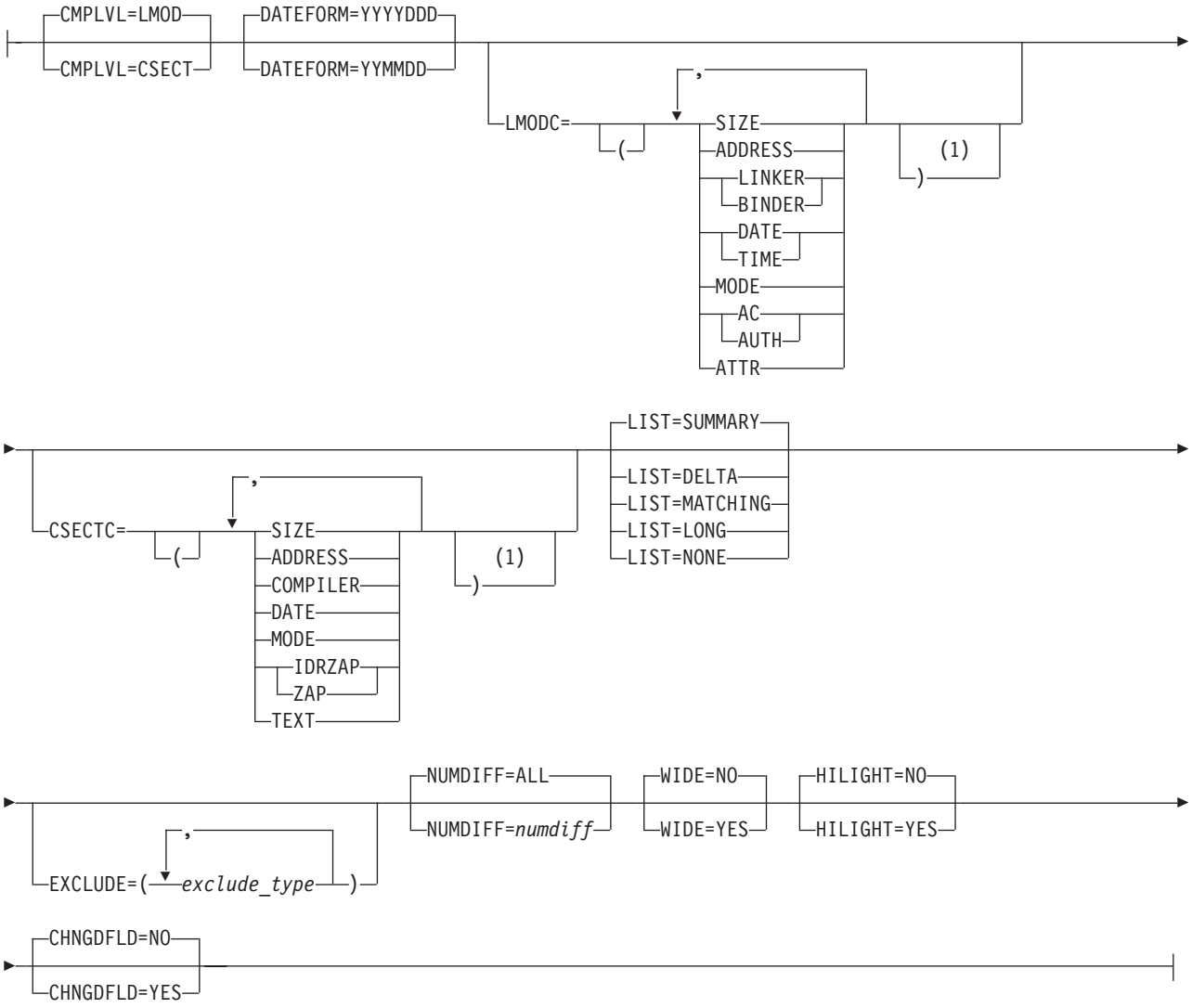


New data set:



Function reference: CLM

Comparison options:



Notes:

1 Provide closing bracket when opening bracket has been used.

Old data set specifications

The "Old" data set can be specified as follows:

DDOLD=*ddold*

Defines a reference to a DD or TSO ALLOC statement for the "Old" load library. The default is DDOLD.

DSNOLD=*dsnold*

Defines the name of the "Old" load library (PDS). If specified, any DD statements provided are not used. The name may include a

member name in parenthesis. If the member is specified here, the associated MEMOLD parameter must be empty. To further describe the data set, use the following:

VOLSEROLD=*volserold*
The VOLUME serial number for a non-cataloged "Old" data set.

MEMOLD=*memold*
The name of a single member in a PDS, or a member name pattern representing one or more members in a load library. You may specify this parameter, or a member name in the DD statement for *ddname*, or specify a member or members in the MEMLIST parameter, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

An * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of *d*, all members in the PDS whose name contains "d" are processed.

A % is a place-holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the PDS with a 4-character name are processed.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the compare. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *memold* parameter of the MEMOLD keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the compare. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *memold* parameter of the MEMOLD keyword.

MEMLIST
Provides a means of selecting members from a load library where no generic name pattern and no member name range has been specified. If the MEMLIST keyword is specified, only those members included in the MEMLIST arguments are compared with the corresponding members in the output data set. Members selected by the MEMBER=*memold* that are not included in the MEMLIST arguments are not compared.

comp_mem
The name of the member to be compared. Generic name masks are not allowed.

SKIPOLD=*skipold*
Number of logical records to be skipped from the beginning of the "Old" data set. The default is 0. Note that the number of logical records describing a load module is equal to the number of CSECTs plus 1 (one record describes the load module itself).

CMPOLD=*cmpold*
Number of records from the "Old" data set to be compared. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are compared. Note that the number of logical records describing a load module is equal to the number of CSECTs plus 1 (one record describes the load module itself).

New data set specifications
The “New” data set can be specified as follows:

DDNEW=*ddnew*
Defines a reference to a DD or TSO ALLOC statement for the "New" load module library. The default is DDNEW.

DSNNEW=*dsnnew*
Defines the name of the "New" load module library (PDS). If specified, any DD statement provided are not used. The name can include a member name in parenthesis. If the member is specified here, the associated MEMNEW parameter must be empty. To further describe the data set, use:

VOLSERNEW=*volsernew*
The VOLUME serial number for a non-cataloged "New" data set.

MEMNEW=*memnew*
The name of a single member in a load library, or a member name pattern representing one or more members in the library. You can specify this parameter, or a member name in the DD statement for *ddname*.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

An * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of *d*, all members in the library whose name contains "d" are processed.

A % is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the library with a 4-character name are processed.

Specification of MEMNEW (or a member in DSNNEW) depends on parameters used in MEMOLD (or member used in DSNOLD). If MEMOLD (member in DSNOLD) specifies one member, MEMNEW (member in DSNNEW) must also point to one member. If MEMOLD (member in DSNOLD) contains a member name pattern, specification of MEMNEW (member in DSNNEW) must use the same pattern or an "*".

SKIPNEW=*skipnew*
Number of logical records to be skipped from the beginning of the 'New' data set. The default is 0. Note that the number of logical records describing a load module is equal to the number of CSECTs plus 1 (one record describes the load module itself).

CMPNEW=*cmpnew*
Number of records from the "New" data set to be compared. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are compared. Note that the number of logical records describing a load module is equal to the number of CSECTs plus 1 (one record describes the load module itself).

Compare options

CMPLVL
Determines the level of the load module comparison.

LMOD
Only information on the load module level is extracted and compared. CSECT information (and differences on CSECT level) are ignored. This results in a less detailed comparison.

CSECT
Information on both the load module and CSECT levels is extracted and compared. This results in a detailed comparison.

LMODC
Determines what information at the load module level is to be included in the compare. The criteria correspond with the load module properties; only those specified are compared and displayed. Each of the options below can be specified in any sequence, enclosed in parenthesis:

SIZE The load module size is compared.

ADDRESS
The entry point address of the load module is compared.

LINKER | BINDER
The version of the linkage editor or the binder used to prepare the load module is compared. LINKER and BINDER are mutually exclusive.

DATE | TIME
The load module link (bind) date and time are compared. DATE and TIME are mutually exclusive.

MODE
The AMODE and RMODE of the load module are compared.

AC | AUTH
The load module authorization code is compared. AC and AUTH are mutually exclusive.

ATTR The load module link (bind) attributes are compared.

CSECTC
Determines what information at the CSECT level is to be included

in the compare. The criteria correspond with the CSECT properties; only those specified are compared and displayed. Each of the options below can be specified in any sequence, enclosed in parenthesis.

Note: If you specify CMPLVL=LMOD, this parameter is ignored.

SIZE The CSECT size is compared.

ADDRESS
The address of the CSECT is compared.

COMPILER
The versions of the language compilers used to compile the CSECT are compared

DATE The date of the CSECT compile is compared.

MODE
The AMODE and RMODE of the CSECT are compared.

IDRZAP | ZAP
The AMSPZAP IDR data is compared. The IDR ZAP data is an extension of the CSECT information, but is formatted into separate records. ISRZAP and ZAP are mutually exclusive.

TEXT The CSECT content is compared. The CSECT content is an extension of the CSECT information, but is formatted into separate, 32-byte records and shown in "memory dump" format (hexadecimal and character).

LIST=SUMMARY
Summary listing.

LIST=DELTA
Delta listing.

LIST=MATCHING
Matching listing.

LIST=LONG
Long listing.

LIST=NONE
No listing.

Listing Options
The following option takes effect if the LIST=LONG parameter is specified:

EXCLUDE=*exclude_type*
The specified compare result types are not reported. *exclude_type* can have the following values:

INSERTED
Excludes inserted records from the report.

DELETED
Excludes deleted records from the report.

CHANGED
Excludes changed records from the report.

MATCHED
Excludes matched records from the report.

The following options take effect if the LIST keyword is not specified (that is, it defaults to SUMMARY) or is set to anything other than NONE:

NUMDIFF=*numdiff*
The number of differences after which the Compare Utility stops processing the data sets.

WIDE=NO
Narrow listing.

WIDE=YES
Wide listing.

HILIGHT=NO
No highlighting of changed fields.

HILIGHT=YES
Highlight changed fields.

CHNGDFLD=NO
Show all fields in the formatted comparison reports.

CHNGDFLD=YES
Show only changed fields in formatted comparison reports. This option has no effect if the 'Wide listing' (WIDE=YES) option has been selected

DATEFORM=YYYYDDD
Reported dates (link and compile dates) shown in YYYY.DDD format.

DATEFORM=YYMMDD
Reported dates (link and compile dates) shown in YY/MM/DD format.

A batch example: Comparison of two versions of a module.

```
//CLM JOB (ACCT),'NAME'
//* COMPARE LOAD MODULES
//*
//FMBAT PROC
//FMBAT EXEC PGM=FILEMGR
//STEPLIB   DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//  PEND
//*
//STPSSEX EXEC FMBAT
//SYSIN DD *
$$FILEM VER
$$FILEM CLM CMPLVL=CSECT,
$$FILEM LIST=DELTA,
$$FILEM WIDE=YES,
$$FILEM HILIGHT=YES,
$$FILEM DSNOLD=FMN.PROD1.LOAD(FMNLOAD),
$$FILEM DSNNEW=FMN.PROD2.LOAD(FMNLOAD),
$$FILEM LMODC=(SIZE,DATE,ATTR),
$$FILEM CSECTC=(SIZE,ADDR,DATE,ZAP)
$$FILEM EOJ
```

Note:

1. Load modules are compared at load module and CSECT level.
2. Load module size, date and time when load module was linked (bound), and link attributes are compared at load module level.
3. CSECT address, size, date of compile and IDR ZAP data are compared at CSECT level.

DP (Disk Print)

Purpose

Print physical disk records.

Usage notes

Use this function to print all records from a physical disk extent in character format or one of the two dump formats. Begin and end positions must be specified as a range of tracks.

Options

You can use the various SET processing options to control the print output:

- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- SET DATAHDR determines whether header information, such as record number, is printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Specify a logical record length to deblock the physical records. If it is not a multiple of the physical record size, the remainder is printed as a short record.

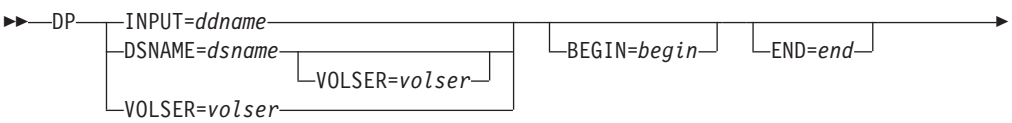
Use the LISTONLY keyword to specify how much information you want to print. You can request the home address and all count, key, and data fields of each track, or only count or key fields. The default is to print all key and data fields.

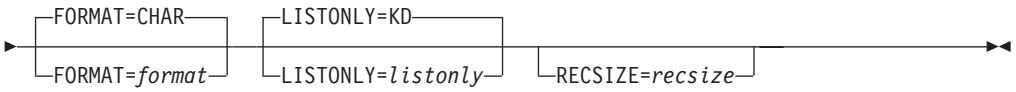
For a CKD disk, use the dump type to specify how much information you want to print. You can request the home address and all count, key, and data fields of each track, or only count or key fields. The default is to print all key and data fields.

Related functions

DRS Locate data on a disk
DSP Print records from a data set

Syntax





begin Start position in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

R*nnnnnn*
To indicate the track number (CKD)

* The first track of the data set (the default).

Depending on the value of the SET CYLHD processing option, you specify either the absolute disk location or the relative location within the data set.

ddname
Refers to a DD or TSO ALLOC statement.

dsname
Name of a disk data set.

end End position in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

R*nnnnnn*
To indicate the track number (CKD)

N*nnnnnnn*
The number of tracks (CKD) to be processed

* The last track of the data set (the default).

Depending on the value of the SET CYLHD processing option, you specify either the absolute disk location or the relative location within the data set.

format
The format of the output:

CHAR
Character format (the default)

HEX Hexadecimal format.

listonly
For a CKD disk, the information you want printed:

KD Key and data fields of each record (the default)

CNT Home address and count fields of each record

CKD Home address and record 0 fields, followed by the count, key, and data fields

KEY Only the key fields.

recsize
Logical record size to be used to deblock each physical record into logical records. The maximum is 65 535. The physical block size does not need to be a multiple of *recsize*. For a CKD disk, only the data field is deblocked.

volser Volume serial number for a non-cataloged data set.

```
//DP  JOB (acct),'name'  Disk Print
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
```

```
//*
//FILEMGR  EXEC FMBAT
//DISK     DD DSN=FMO.H0GB210.SFMNMOD1,DISP=SHR
//SYSIN    DD *
$$FILEM SET CYLHD=RELATIVE
$$FILEM DP  INPUT=DISK,BEGIN=R0,END=N10
$$FILEM EOJ
/*
```

DRS (Disk Record Scan)

Purpose

Locate data on a disk.

Usage notes

Use this function to locate physical disk records containing data that meet your search criteria. The scan range must be specified as a range of tracks.

You can search for a specific string or for end-of-file records. The search string can contain up to 50 characters and can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in quotation marks. The string can contain blanks and commas and must match exactly (including case).
- A hexadecimal string, for example X'04'.

File Manager prints all matching records in hexadecimal dump format and lists on the terminal the record location and the position of the string.

Options

Specify a record length to deblock the physical records before searching. The search can apply to the key or data part of the record. You can request searching everywhere in the data, from a position you specify, or only at the specified offset.

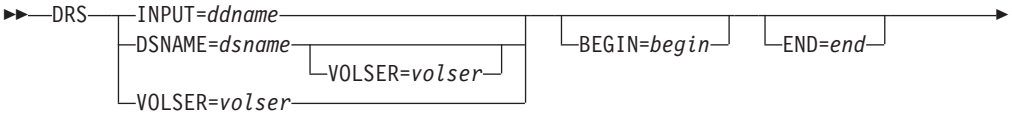
You can use the various SET processing options to control the print output:

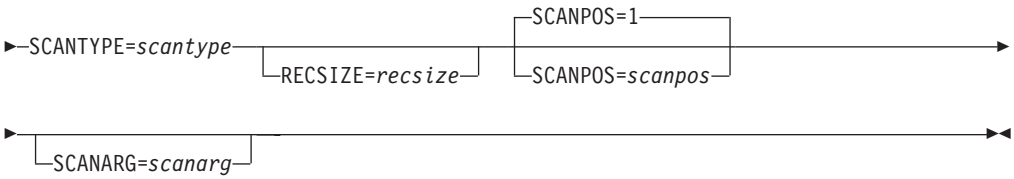
- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Related functions

DP Print physical disk records
FCH Find or change data

Syntax





begin Start position in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

R*nnnnnn*
To indicate the track number (CKD)

* The first track of the data set (the default).

Depending on the value of the SET CYLHD processing option, you specify either the absolute disk location or the relative location within the data set.

ddname
Refers to a DD or TSO ALLOC statement.

dsname
Name of a disk data set.

end End position in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

R*nnnnnn*
To indicate the track number (CKD)

N*nnnnnnn*
The number of tracks (CKD) to be processed

* The last track of the data set (the default).

Depending on the value of the SET CYLHD processing option, you specify either the absolute disk location or the relative location within the data set.

recsize
Logical record size to be used to deblock each physical record into logical records. The maximum is 65 535. The physical block size does not need to be a multiple of *recsize*.

scanarg
The string to be searched for. The string can contain up to 50 characters and can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
- X followed by a hexadecimal string enclosed in quotation marks (X'C1C2'), or a hexadecimal string enclosed in quotation marks followed by `X` ('C1C2'X).

The *scanarg* parameter is required unless *scantype* is E.

scanpos
Offset (byte position) where the search begins within a record. The default is 1.

scantype
One of the following:

D Scans data fields *only* at the position in the record specified by *scanpos*

DP Scans data fields *starting* at the position in the record specified by *scanpos*

K Scans key fields *only* at the position in the record specified by *scanpos* (CKD disks only)

KP Scans key fields *starting* at the position in the record specified by *scanpos* (CKD only)

E Scans for end-of-file records.

volser Volume serial number for a non-cataloged data set.

```
//DRS JOB (acct),'name'  Disk Record Scan
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//DELETE   EXEC PGM=IEFBR14
//SYSUT2   DD DSN=FMNUSER.TEMP.SEARCH.FILE,
//            DISP=(MOD,DELETE),
//            UNIT=SYSALLDA,
//            SPACE=(TRK,(3,1),RLSE),
//            RECFM=FB,LRECL=80,BLKSIZE=24000
//*
//MAKEFILE EXEC PGM=IEBGENER
//SYSIN    DD *
/*
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD *
Andrew Apple
Ted Dexter
Grant Smith
Keith Sampson
Graham Prestcott
John Laws
Bill McCork
Liz Childs
/*
//SYSUT2   DD DSN=FMNUSER.TEMP.SEARCH.FILE,
//            DISP=(,CATLG),
//            UNIT=SYSALLDA,
//            SPACE=(TRK,(3,1),RLSE),
//            RECFM=FB,LRECL=80,BLKSIZE=24000
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//DISK     DD DSN=FMNUSER.TEMP.SEARCH.FILE,
//            DISP=SHR
//SYSIN    DD *
$$FILEM DRS INPUT=DISK,BEGIN=*,END=*,
$$FILEM  SCANTYPE=DP,SCANARG='Laws'
$$FILEM EOJ
/*
```

DSB (Data Set Browse)

Purpose

The DSB function displays the Data Set Browse or Browse Entry panel, in

online mode. This function can be invoked from a REXX program or a TSO clist or entered on an ISPF Command line, but **cannot** be used in batch jobs as it is an interactive function.

Usage notes

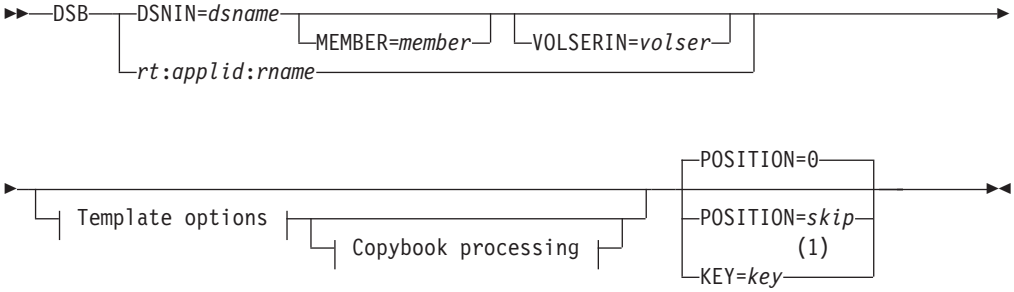
When invoked from an ISPF Command line without a parameter, the Data Set Browse Entry panel is displayed in interactive mode and you can specify a valid data set for browsing. If you include a valid data set name as a parameter, then you bypass the Entry panel, and go directly to browsing the specified data set.

When invoked from a REXX procedure or TSO list, you must supply the data set name.

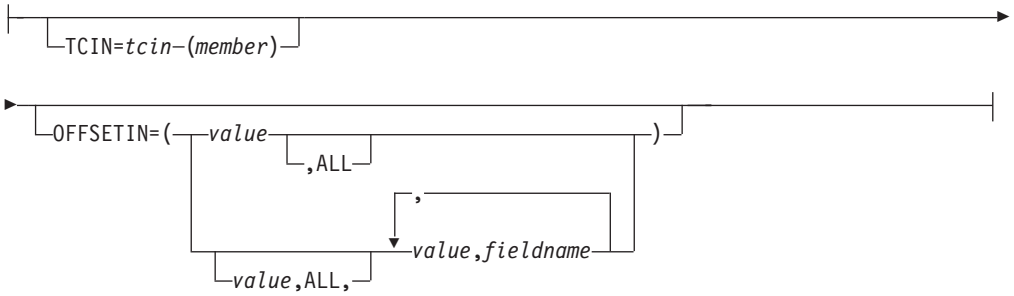
Related functions

DSE Display Data Set Edit or Edit Entry panel
DSEB Update disk data set records
DSU Update disk data set records

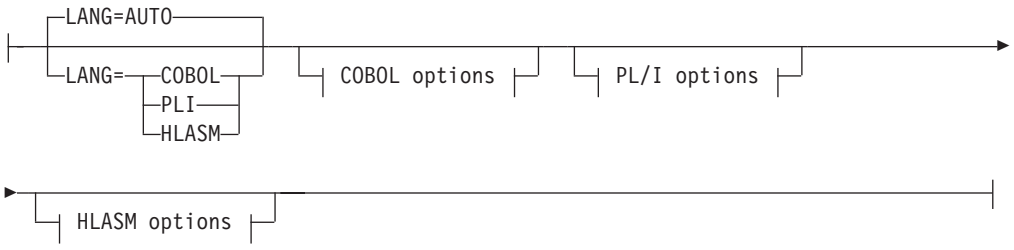
Syntax



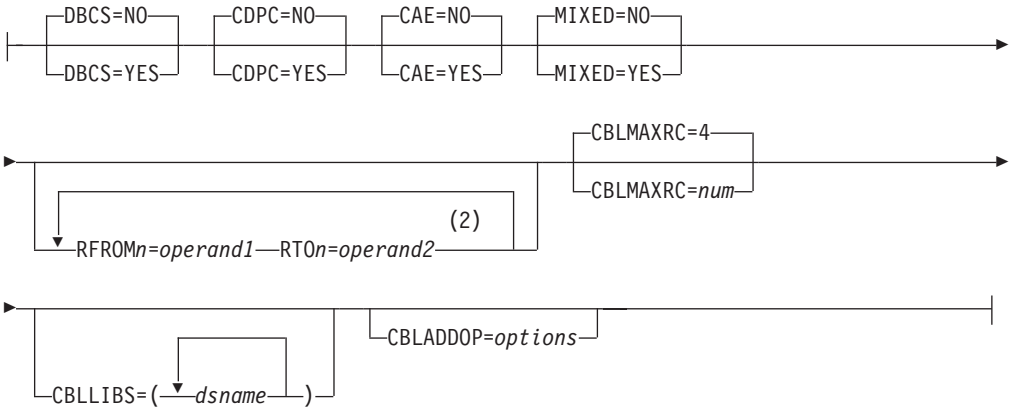
Template options:



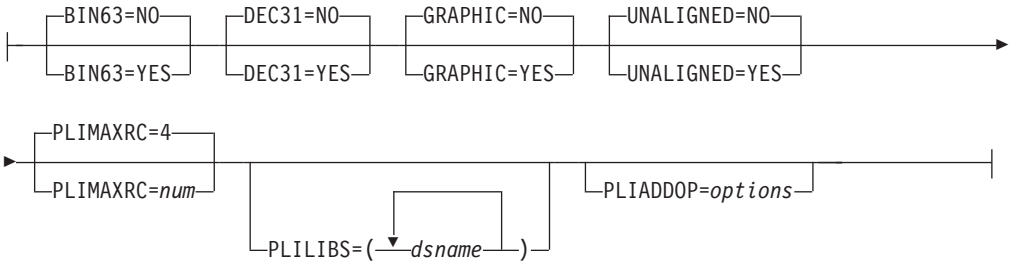
Copybook processing:



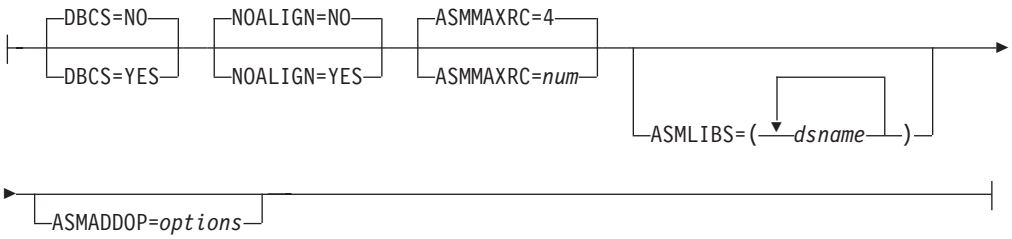
COBOL options:



PL/I options:



HLASM options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

DSNIN=*dsname*

Defines the name of the data set to be browsed. If specified, any DD statement provided are not used. The name can include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty.

Must be specified when $DSB is invoked from a REXX procedure or TSO clist.

Can be omitted when DSB is entered on a command line. If specified, you bypass the Entry panel, and go directly to the Data Set Browse panel.

When omitted, the Data Set Browse Entry panel is displayed in interactive mode and you can specify a valid data set for browsing.

You can further describe this data set, as follows:

MEMBER=*member*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the data set is a PDS, you must specify this parameter.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%`, all members in the PDS whose name is four characters in length are processed.

MEMBER=*member* is ignored if the data set is not a PDS.

rt:*applid*:*rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, browsing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

TCIN=*tcin***(***member***)**
PDS and member name of the copybook or template that describes the record structure of your data set.

OFFSETIN
The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'===='` as *operand2*. This would be specified as:

```
RFROM1==:==,RTO1====,
```

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

- "Invoking File Manager panels from outside File Manager" on page 13
- Chapter 12, "Introduction to programming with File Manager functions," on page 397
- "Invoking File Manager panels from REXX procedures or TSO clists" on page 400


```
FILEMGR "$DSB DSNIN='FMNUSER.TEST.KSDS1',KEY=000100"
FILEMGR DSB
```

DSC (Data Set Copy)

Purpose
Use the DSC function to copy data from any supported data set to any other supported data set. The function's performance when copying PDS(E)es greatly depends on the available storage: larger regions generally result in better performance.

Usage notes

- You can select the records to be copied using:
  - Member name selection criteria
  - Date created selection criteria

  - Date last modified selection criteria
  - User ID selection criteria
  - The start key (VSAM only)
  - The skip field
  - The copy count field
  - A conditional expression defined in the *from* template
- Change data set attributes. File Manager can copy records where the input and output data sets have different record formats, record lengths, or block sizes. The copy process truncates or pads records appropriately. Specify the pad character in the PAD field of the SET function. For details see "SET (Set Processing Options)" on page 1047.
- Copy from field to field. Using both a "From" and a "To" template lets you copy selected fields, change the size and type of fields, and create new fields in the output data set. For details, see "Copying data sets" on page 253.
- Copy to output in external format. The "From" template defines the traditional format of a dataset, but also determines a natural character representation of the data. The result of the generation is an output dataset containing a copy of the input data in an external format (such as XML). For details, see "Generating data in external format - XML representation" on page 263.
- Copy concatenated data sets with like or unlike attributes. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for processing. Such invocation may fail as DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable the DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.
- Change ISPF packed format. File Manager can unpack existing packed members or sequential data sets, or can write members or sequential data sets in ISPF packed format.
- These changes are visible in the printed DSC BATCH processing report:
  - Member names are printed as specified on the CPYMBR list (when used).
  - Whenever a member name, its alias or new name (prompt value) contains unprintable characters, an additional line of output is printed below the regular output containing the hexadecimal values of the respective member names.
  - Whenever a member was located in the input library and appeared on the CPYMBR list but was not selected for processing because of the member mask or advanced member selection criteria, then it is shown in the processing report as "Not selected" (as opposed to "Not found") and be counted in the "not copied" category (as opposed to "in error").
  - For example, with these control cards:

    ```
    $$FILEM MEMBER=X'5C22',
    $$FILEM CPYMBR=(C'allocplx',
    $$FILEM         x'8289879784a222',
    $$FILEM         autotest,
    $$FILEM         X'84A282')
    ```

    One would see this output:

```
                                                          Member Copy Report
Member         Newname    Alias      Status
-------------------------------------------------------------------------
allocplx                             Not selected
bigpds                               Replaced
X'8289879784A222'
AUTOTEST                             Not found
dsb                                  Not selected
FMN4688I 0 member(s) copied; 1 replaced; 2 not copied; 1 in error
```

- **Member names containing lowercase or unprintable characters:**
  - Member names specified with the CPYMBR, MEMBER, MEMSTART, MEMEND, or MEMOUT keywords may contain lowercase or unprintable characters.
  - To specify a member name containing lowercase or mixedcase characters, use the character literal form of the name surrounded by quotes and preceded with character C. For example, `C'aBc'`.
  - To specify a member name containing unprintable characters, use the hexadecimal literal form of the name surrounded by quotes and preceded with character X. For example, X'81C283'. Mask characters (their hexadecimal value) may be included within the string.

Note: File Manager supports the copying of Load Modules, when the following conditions are met:

- Your input and output data sets are PDS(E)es.
- Your TSO environment is active (and you can use the TSO authorized program services), or you are running File Manager as program-authorized.
- You have not specified a REXX user procedure.
- You have not specified Start key, Skip or Copy counts.
- You are not using templates.
- You do not request member record counts.

Performance tips

- See "General tips about performance when you use File Manager functions" on page 833. The comments about File Manager using DFSORT technology when performing sequential file I/O are important to DSC performance.
- When you are using DSC to copy members of a PDS(E):
  - One DSC default is STATS=ON, which causes the ISPF statistics for each copied member to be updated. This can significantly increase I/O (EXCP) and CPU utilization. To improve performance, consider using STATS=OFF.
  - File Manager attempts to use IEBCOPY or an equivalent product to copy members if it can. File Manager using IEBCOPY can significantly reduce I/O and CPU requirements, compared to File Manager not using IEBCOPY. If any File Manager processing of individual records is required, it cannot use IEBCOPY. For example, File Manager cannot use IEBCOPY if:
    - A proc (PROC=) is used.
    - A template or copybook is used.
    - Record counts are requested (RECCOUNTS=YES).
- File Manager does not use IEBCOPY when processing members of a PDS(E) when it detects any member names containing unprintable or

lowercase characters since IEBCOPY is not capable of processing such member names. This may negatively affect the performance of the DSC operation.

Options

When you specify the PROC option, you are supplying a DFSORT or REXX procedure that controls the selection and formatting used during the copy function. For more information, see the *proc* parameter below.

Return codes

The default return codes from the DSC function have the following modified meanings:

1 No records copied for some of multiple members.

2 No records copied for any of multiple members.

3 REXX member selection is in effect but the procedure encountered a RETURN DROP, STOP or STOP IMMEDIATE string. This has been treated as a RETURN string with no arguments.

OR

REXX member selection is NOT in effect but the procedure encountered a RETURN DROP MEMBER, RETURN PROCESS MEMBER string. This has been treated as a RETURN string with no arguments.

4 No records copied because no records selected (for single member or data set)

4 No records copied because no members to process

4 No records copied because input empty

4 No records copied because member exists and “no replace” option specified

4 Input data set or member was skipped because it is in ISPF Packed Data format and the “PACK=SKIP” option was specified

8 REXX non-syntax error encountered while processing records

8 Line generated in an external format is greater than output record size.

16 No records copied because input and output physically the same (not applicable to a PDS member)

16 Program Object specified - this is not supported

16 Output data set or member in use

16 Data set or member open error

16 Data set or member not found

16 Other input or output error occurred

16 Insufficient storage available

16 DSC abended

16 Input data appears ISPF packed but is not valid.

16 Other serious error that stops processing occurred

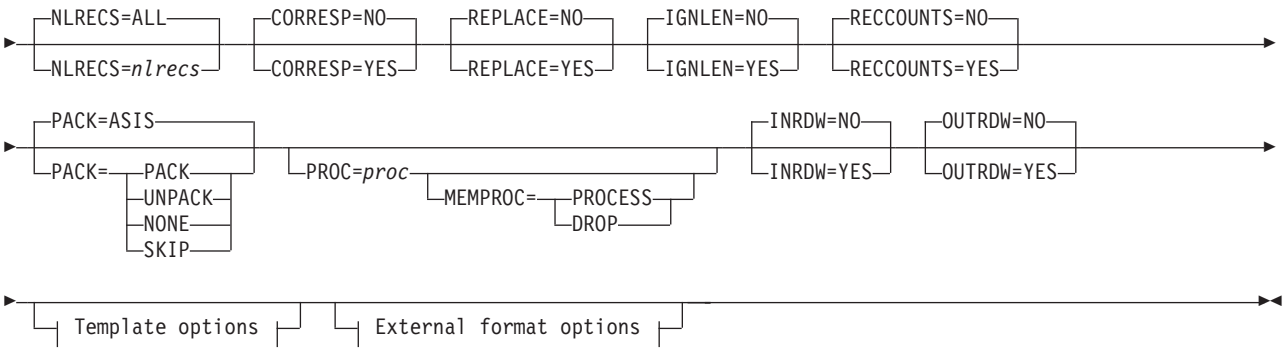
Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Related functions

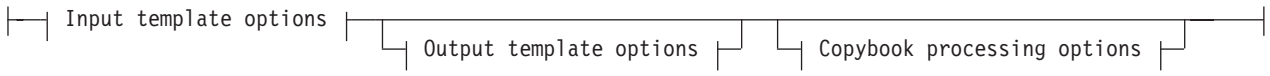
OS Backup objects from an OAM database to a data set
OV Backup objects from an OAM database to a VSAM data set
TS Copy tape data to a data set
SO Copy a data set to an object database
ST Copy a data set to tape
VO Copy VSAM data to an object database

Syntax

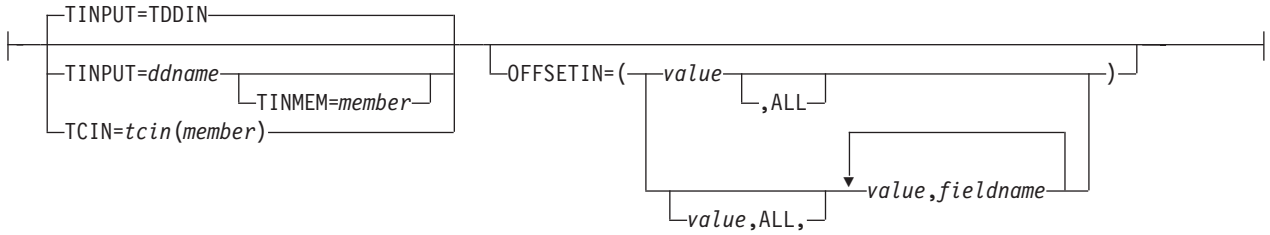
Function reference: DSC



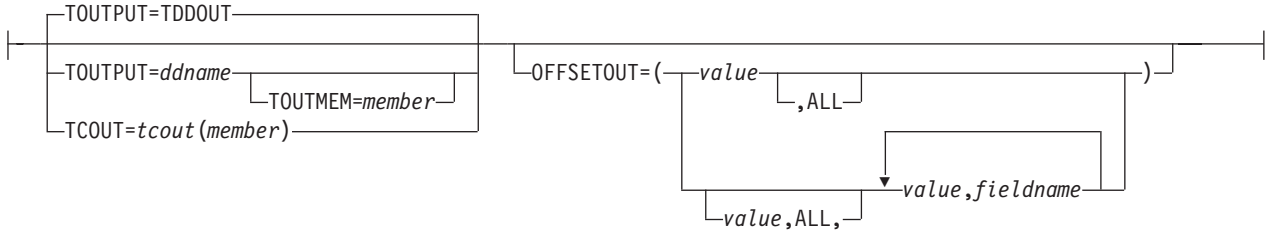
Template options:



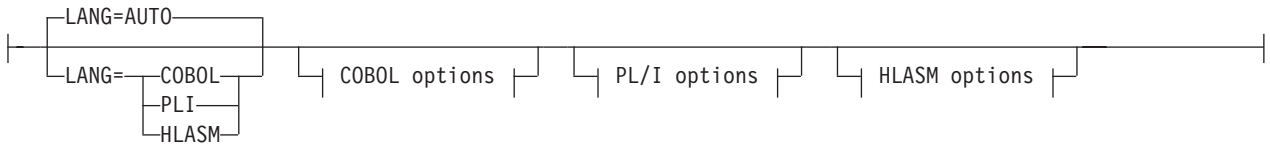
Input template options:



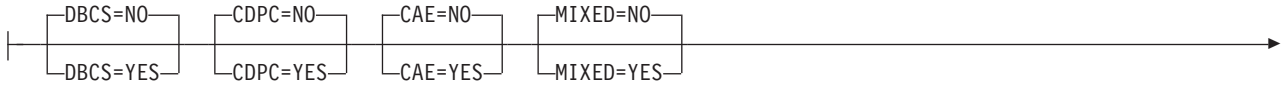
Output template options:

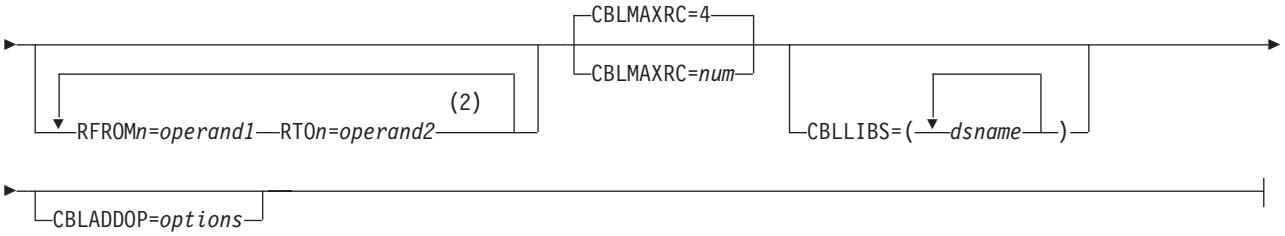


Copybook processing options:

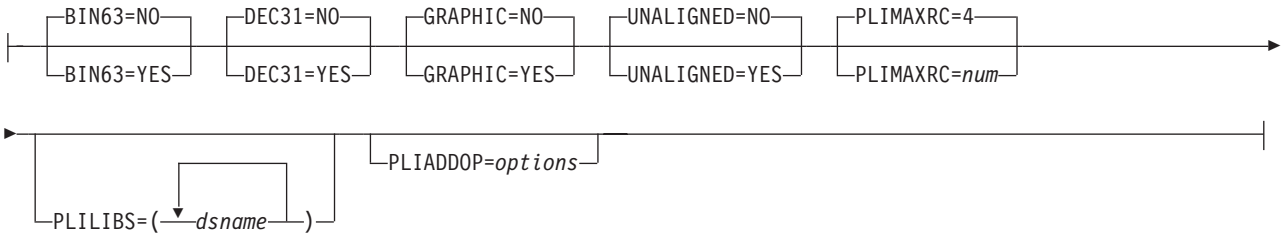


COBOL options:

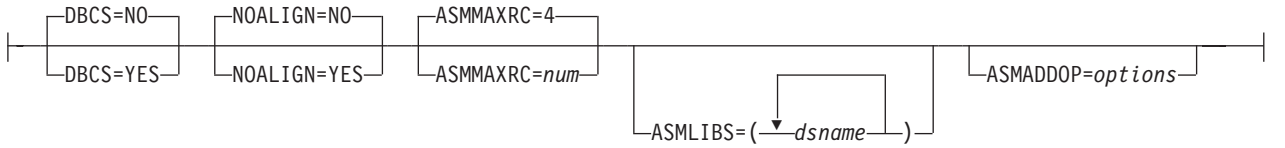




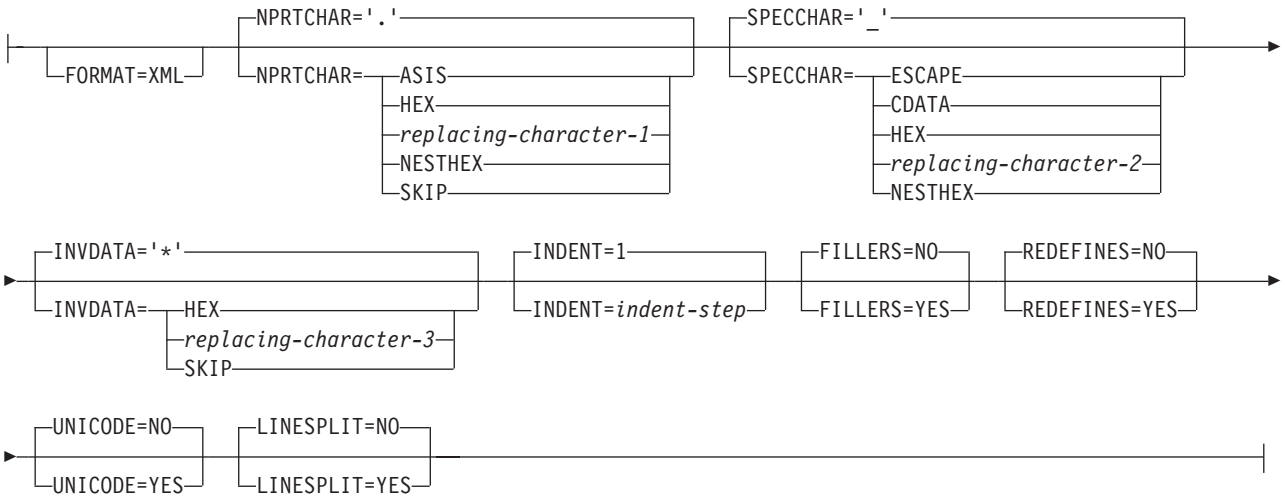
PL/I options:



HLASM options:



External format options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

INPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the "From" data set or HFS file. The default is `DDIN`.

DSNIN=*dsname*
Defines the name of the "From" data set or an absolute path to the "From" HFS file (directory). If specified, any DD statement provided are not used. The name can include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line. You can further describe this data set, as follows:

VOLSERIN=*volser*
Volume serial number for a non-cataloged "From" data set.

MEMBER=*member1*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the input data set is a PDS(E), you may specify this parameter, or a member name in the DD statement for *ddname*, or specify a member or members in the CPYMBR parameter, or specify a range of member names via the MEMSTART and/or MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member1 is ignored if the data set is not a PDS.

Note: See 875.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the copy. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

Note: See 875.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the copy. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

Note: See 875.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

manager
The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuename
The queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

Generic names are not supported for MQ when running in batch.

*rt***:***applid***:***rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the copy.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the copy.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00'). If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

CPYMBR
Provides a means of selecting input members from a PDS(E) where no generic name pattern and no member name range has been specified. Also provides a means for renaming the selected members as they are copied to the output data set. If the CPYMBR keyword is specified, only those members included in the CPYMBR arguments are copied to the output data set. Members selected by the MEMBER=*member1* that are not included in the CPYMBR arguments are not copied.

The CPYMBR arguments also refine the member list specified in the MEMBER=*member1* parameter. If the MEMBER keyword is *not* specified, it is assumed to be MEMBER=*, and all members named in the CPYMBR list are processed. However, if the MEMBER keyword is specified, for example as MEMBER=TEST*, the members included in the CPYMBR list are selected from the TEST* subset of members. Any members named in the CPYMBR arguments that do not match the mask given in the MEMBER parameter are not copied.

from_mem
The name of the member to be copied. Generic name masks are allowed.

to_mem
The name of the member after it has been copied to the output data set. If unspecified, the output member is not renamed.

Note: See 875.

BINRECIN=*binrecin-len*
Specifies the record length used for processing the “From” HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

USEIOXIN
Specifies whether to invoke a user I/O exit, to process the input data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the input data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXIN
Specifies the name of the user I/O exit used for the input data set. There are no restrictions on the programming language that you can use to write

an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXIN=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXIN has been set to YES and no installation default has been provided, you must specify IOXIN=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

EXCLUSIVE

Note: This option is supported for backward compatibility only.

Use the new DIRINTEGR option.
Determines the disposition of the From (input) data set.

NO Default. The data set is allocated with DISP=SHR, so that other users can obtain concurrent access to a PDS or PDSE during execution of DSC.

YES The data set is allocated with DISP=OLD, preventing concurrent access to the PDS or PDSE.

Note: If you pre-allocate the input data set with DISP=SHR and then specify,EXCLUSIVE=YES in batch, I/O errors might occur during concurrent access to the data.

DIRINTEGR
Specifies whether to invoke a user I/O exit to process the input data set.

NO Default. File Manager uses a faster PDS(E) directory processing method. This may cause I/O errors when multiple users are concurrently updating the directory of the data set being processed.

YES File Manager uses safer, but slower, PDS(E) directory processing method. This method allows for safe concurrent updates of the PDS(E) directory by multiple users.

INCLUDEMD
This parameter determines what happens to the descriptor for an Websphere MQ message.

YES The descriptor is added as a prefix as the data is copied.

NO Just the message data is copied. This is the default.

This parameter applies only when the input describes an MQ queue.

JCL=NO
Treat the data set as a non-JCL data set.

JCL=YES
Indicates the file or PDS(E) members being copied contain JCL, the syntax of which is to be maintained during the copy process.

This option is only available when the **PROC** option is selected as changes can only be made to the input data by means of REXX processing.

This option is ignored if a template is being used.

Use of this option may affect the performance of the copy due to the overhead of verifying the syntax of the JCL.

OUTPUT=ddname
Defines a reference to a DD or TSO ALLOC statement for the "To" data set or HFS file. The default is `DDOUT`.

DSNOUT=*dsname*
Defines the name of the "To" data set or an absolute path to the "To" HFS file (directory). If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. You can further describe this data set, as follows:

(*member2***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name. An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it overmore than one line.

VOLSEROUT=*volser*
Volume serial number for a new or non-cataloged "To" data set.

BINRECOUT=*binrecout-len*
Specifies the record length used for processing the "To" HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

Note: See 875.

USEIOXOUT
Specifies whether to invoke a user I/O exit, to process the output data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the output data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

MEMOUT=*mask*
Where a number of input members have been specified, you can specify a member name pattern for the output members, allowing you to rename your members as they are copied. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple

characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

MQOPEN
This parameter determines if the Websphere MQ queue is opened in browse or share mode and the messages from that queue read in a destructive manner.

BROWSE
The Websphere MQ queue is opened in browse mode. Messages are not removed from the queue as they are read. This is the default.

SHARE
The Websphere MQ queue is opened in share mode. Messages are removed from the queue as they are read (the queue is read in a destructive manner).

This applies only when the input describes an MQ queue.

IOXOUT
Specifies the name of the user I/O exit used for the output data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXOUT=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXOUT has been set to YES and no installation default has been provided, you must specify IOXOUT=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

STATS=ON
Default. This updates the ISPF statistics (if already present) when a PDS or PDSE member has been changed.

STATS=OFF
The ISPF statistics are not updated when a PDS or PDSE member has been changed.

STATS=FORCE
The ISPF statistics that exist for members being processed are always updated and statistics for a member that previously did not have statistics are created.

DISP Determines the disposition of the To (output) data set. Specify OLD or MOD.

OLD Writes input records to the existing output data set, starting from the beginning.

MOD Default. Appends the input records to the end of the existing output data set.

Note: MOD is not available for PDS(E) member processing.

Note: SMS might modify the allocation of new data sets on your system. For details, contact your SMS Administrator.

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

SMPLINCL=*sample_include*
Number of physical records to be included in a repeating sample from a data set. Both SMPLINCL and SMPLSKP keywords must have non-zero values for sampling to take effect. Range: 0–9999999.

SMPLSKP=*sample_skip*
Number of physical records to be skipped in a repeating sample from a data set. Both SMPLINCL and SMPLSKP keywords must have non-zero values for sampling to take effect. Range: 0–9999999.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

NLRECS
Number of records to be copied or ALL.

ALL If you specify ALL or omit the parameter, all the remaining records are copied.

nlrecs The maximum number is 999 999 999.

When you are coding a REXX procedure and NLRECS is specified, then this can affect the number of records presented to the REXX procedure. NLRECS only applies to the number of records written to the primary output data set. It does not apply to records written in the REXX procedure with the WRITE() function.

CORRESP
Specifies whether or not File Manager maps output fields to input fields with the corresponding name. The default is NO.

NO Instructs File Manager to use the existing field mapping in the TCOUT member. If the TCOUT member is a copybook, or no field mapping is supplied, then File Manager ignores this option and performs a corresponding copy (as if you had specified CORRESP=YES).

YES Instructs File Manager to map output fields to input fields with the corresponding name.

If you want to use existing mapping in the "To" template, specify CORRESP=NO.

REPLACE
Specifies whether or not File Manager replaces like-named members in an output partitioned data set. The default is NO.

NO Like-named members in the output partitioned data set are not replaced.

YES Like-named members in the output partitioned data set are replaced.

IGNLEN
Specifies whether or not File Manager ignores length mismatches when selecting records for processing.

NO Do not ignore length mismatches. Records that are shorter than the matching structure length in the template are not selected for processing.

YES Use this option to ignore length mismatches.

When a field in the "From" template spans or is beyond the copied record's boundary, the corresponding field on the output record is initialized (since there is no data available from the from field). The exception is alphanumeric fields, where the portion of the field that exists on the input record is copied (a partial copy) and the remainder of the output field is padded with blanks.

RECCOUNTS
Controls whether or not the count of records for copied PDS(E) members and sequential/VSAM data sets is printed in the processing listing in batch.

NO Record counts are not reported.

YES Record counts are reported.

Note: This option affects PDS(E) processing ONLY. For sequential/VSAM data sets, the record counts are always provided. When the option is selected, it prevents the use of IEBCOPY for PDS(E) processing, which may affect the copy performance.

PACK Determines if File Manager should detect if the input data is in ISPF packed format and specifies if the output data is to be written in ISPF packed format. This keyword is ignored when processing VSAM data sets. When an I/O exit has been specified for either the input or output data set (or both), the only valid option is PACK=NONE.

ASIS Instructs File Manager to write the output in ISPF Packed format only if the input is in ISPF packed format.

PACK Instructs File Manager to write the output in ISPF packed format regardless of the input format.

UNPACK
Instructs File Manager to write the output without ISPF packing, regardless of the input format.

NONE
Instructs File Manager not to determine if the input data set is in ISPF packed format and writes the output records as they are read from the input data set (after any enhanced processing).

SKIP Instructs File Manager to determine if the input data set is in ISPF packed format and if so, to skip the copy processing.

PROC=*proc*
Member name of a REXX procedure that you want to use to process each record before it is copied, or an asterisk (*) to indicate the procedure is inline. If you specify a member name, you must define an FMNEXEC ddname that identifies the PDS containing the member. If you specify *, the procedure is read from SYSIN immediately following the control statement for the current function. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

For more information about using DFSORT or REXX procedures to process records before they are copied, see Chapter 13, "Enhancing File Manager processing," on page 405.

If PROC=*proc* is specified, you can then choose to include a MEMPROC parameter:

MEMPROC
Specifies that REXX member selection is in effect. Records are read from the input member and then cached in memory until a decision is made, within the REXX procedure, on whether the member is to be copied or dropped. Once the decision has been made, the entire member is either copied or dropped, depending upon the RETURN string specified in the REXX procedure.

If the entire member is processed without encountering a RETURN DROP MEMBER or RETURN PROCESS MEMBER string, the member is processed according to the action specified by the parameter specified for MEMPROC. These are:

PROCESS
The member is to be included in the copy. The member is copied intact, subject to any specified template processing, which is performed before the user REXX proc is invoked.

This is the default action, if no parameter is specified with the MEMPROC keyword.

DROP The member is to be excluded from the copy. Processing continues with the next member.

INRDW
Controls whether or not to adjust the input start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the input start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the input record.

OUTRDW

Controls whether or not to adjust the output start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the output start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the output record.

Template options

The template options define which templates (if any) are used to describe the record structure in the "From" and "To" data sets, and how File Manager processes those templates.

TINPUT=*ddname*

Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your input data. The default is TDDIN.

If you specify a concatenated DD, then you must provide the member name, *member*.

Note: When you specify concatenated data sets in the template DD statement, and these data sets are vendor-managed copybook libraries, a maximum of 20 data sets are supported.

TINMEM=*member*

The name of the copybook or template member in the datasets identified by the TINPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCIN parameter is specified.

TCIN=*tcin***(***member***)**

PDS and member name of the copybook or template that describes the record structure of your input data.

OFFSETIN

The length of the 01 field in the "From" template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the "From" template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 within the "From" template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the "From" template.

TOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your output data. The default is TDDOUT.

If you specify a concatenated DD, then you must provide the member name, *member*.

TOUTMEM=*member*
The name of the copybook or template member in the datasets identified by the TOUTPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCOUT parameter is specified.

TCOUT=*tcout***(***member***)**
PDS and member name of the copybook or template that describes the record structure of your output data.

OFFSETOUT
The length of the 01 field in the "To" template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the "To" template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the "To" template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is applied. The default is the first Level 01 field in the "To" template.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second

compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

External format processing

When generating the output in an external format, File Manager uses the following options:

FORMAT

Specifies the external format to be used for output.

XML Indicates that XML format is used.

NPRTCHAR

Determines how non-printable characters are to be represented in the output.

'.' (dot)

Default. Each non-printable character is replaced with ".".

ASIS Non-printable characters appear unchanged in the output.

HEX Any non-printable character is converted into its hexadecimal representation.

'*replacing-character-1*'

Each non-printable character is replaced with the replacing-character-1. The set of allowable characters is limited to printable characters with exception of special characters. You may specify:

char A character, such as "?".

C'*char***'** A character used without case translation.

X'*cc***'** A character defined by its hexadecimal value.

NESTHEX

Each string of consecutive non-printable characters will be nested into content of element as X'hex-representation-of-string- of-non-printable-characters'

SKIP Any non-printable character causes the value to be skipped (data is represented by start and end tags, without any content).

SPECCHAR

Determines how special characters are to be represented in the XML output.

'_' (underscore)

Default. Each special character is replaced with "_".

ESCAPE

Special characters are converted into escaped strings:
">" for ">"
"<" for "<"
"'" for "'"
""" for "'"
"&" for "&"

CDATA

Unchanged string containing special characters are enclosed into the CDATA section.

HEX Any special character will cause the XML value to be converted into its hexadecimal representation.

'*replacing-character-2***'**
Each special character is replaced with the replacing-character-2. The set of allowable characters is limited to printable characters with exception of special characters. You may use:

char A character, such as "?".

C'*char***'** A character used without case translation.

X'*cc***'** A character defined by its hexadecimal value.

NESTHEX
Each string of consecutive special characters is nested into content of element as X'hex-representation-of-string-of-special-characters'

INVDATA
Determines how invalid data is to be represented in the output.

'*' (asterisk)
Default. Invalid data is represented by string of "*" with a length equal to the assumed length of the output value.

HEX Invalid data causes the output value to be a hexadecimal representation of the input value.

'*replacing-character-3***'**
Invalid data is represented by a string of *replacing-character-3* with a length equal to the assumed length of the output value. The set of allowable characters is limited to printable characters with exception of special characters. You can use:

char A character, such as "?".

C'*char***'** A character used without case translation.

X'*cc***'** A character defined by its hexadecimal value.

SKIP Invalid data is skipped (data is represented by start and end tags, without any content).

INDENT
Specifies the number of blanks used to indent each level of XML tag corresponding to the nested level in the template or copybook.

1 Default. Each nested level causes an increase in indentation of each XML level by one blank.

indent-step
Any value from 0 to 9. INDENT=0 will force no indentation. Each positive number will cause an increase in indentation of each XML level by this number of blanks.

FILLERS
Indicates whether fillers (unnamed data elements) are to be included into the output or not.

NO Default. Fillers are ignored (not represented in the output).

YES Fillers are treated as named data elements and represented in the output.

REDEFINES
Indicates whether data elements redefining other data elements are to be included into the output or not.

NO Default. Redefines are ignored (not represented in the output).

YES Redefines are treated as other data elements and represented in the output.

UNICODE
Indicates whether the output is to be converted to Unicode or not.

NO Default. The output is not converted.

YES The output is converted to Unicode.

LINESPLIT
Indicates, whether the output lines resulting from processing an input record are spanned contiguously over multiple output records, or each output line must be included as the only line in an output record.

NO Default. Output line is contained, as the only output line, in one output record.

YES Output records are cut independently of external formatting. An output line can span multiple output records and not necessarily start from the beginning of the record. However, output representation of each input record starts from the new output record.

Note: You cannot specify different options for compiling “From” and “To” copybooks; the same copybook options are used for both.

```
//DSC JOB (acct),'name'
//* Copy data set to data set
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//STPSSEX     EXEC FMBAT
//SYS1INP  DD DISP=SHR,
//            DSN=SYS1.PARMLIB(LNKLST00)
//SYSIN    DD *
$$FILEM VER
$$FILEM DSC INPUT=SYS1INP,
$$FILEM     DSNOUT=FMNUSER.TEMP.LINKLIST,
$$FILEM     TCIN=FMNUSER.FMN.TEMPLATE(ODOTWO01)
$$FILEM EOJ
/*
```

This example copies all members of the input PDS ('USERID.PLXIN') that match the input mask 'FMNE*' to the output PDS('USERID.PLXOUT'), renaming them using the rename mask JBG*. The members FMNEDIT, FMNEDIT1, FMNEDIT2,

FMNEDIT3, FMNEDIT4, FMNEDIT5 and FMNEDIT6 are copied and renamed as JBGEDIT, JBGEDIT1, JBGEDIT2, JBGEDIT3, JBGEDIT4, JBGEDIT5 and JBGEDIT6 respectively.

```
//SYSIN DD *
$$FILEM DSC DSNIN=USERID.PLXIN,
$$FILEM MEMBER=FMNEDIT*,
$$FILEM DISP=MOD,
$$FILEM DSNOUT=USERID.PLXOUT,
$$FILEM MEMOUT=JBG*
```

This example shows copying PDS(E)s using a TSO environment (required for load module processing when File Manager is not APF-authorized and recommended for better performance with PDS(E)s).

```
//FMBAT EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
CALL *(FMNMAIN)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM DSC DSNIN=TEST.PDS,
$$FILEM MEMBER=*,
$$FILEM REPLACE=YES,
$$FILEM DISP=OLD,
$$FILEM DSNOUT=TEST2.PDS
```

DSE (Data Set Edit)

Purpose

The DSE function displays the Data Set Edit or Edit Entry panel, in online mode. This function can be invoked from a REXX program or a TSO clist or entered on an ISPF Command line, but **cannot** be used in batch jobs as it is an interactive function.

Usage notes

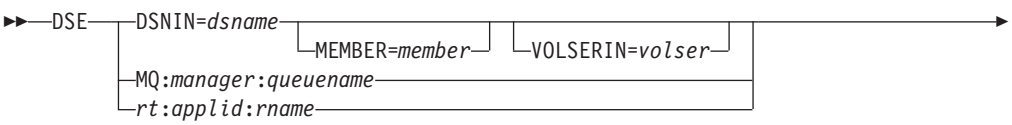
When invoked from an ISPF Command line without a parameter, the Data Set Edit Entry panel is displayed in interactive mode and you can specify a valid data set for browsing. If you include a valid data set name as a parameter, then you bypass the Entry panel, and go directly to editing the specified data set.

When invoked from a REXX procedure or TSO list, you must supply the data set name.

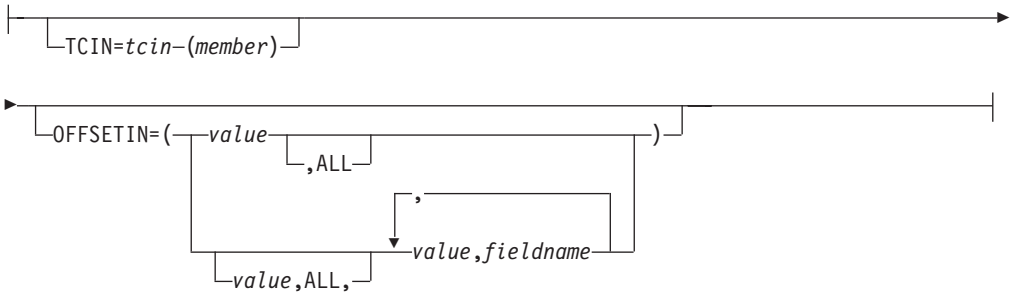
Related functions

DSB Display Data Set Browse or Browse Entry panel
DSEB Update disk data set records
DSU Update disk data set records
DSV Display the Data Set View or View Entry panel

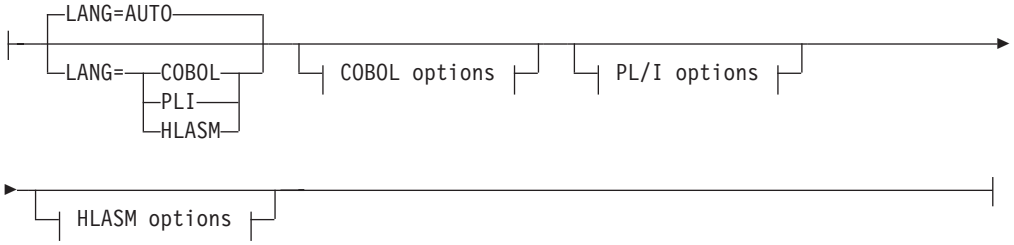
Syntax



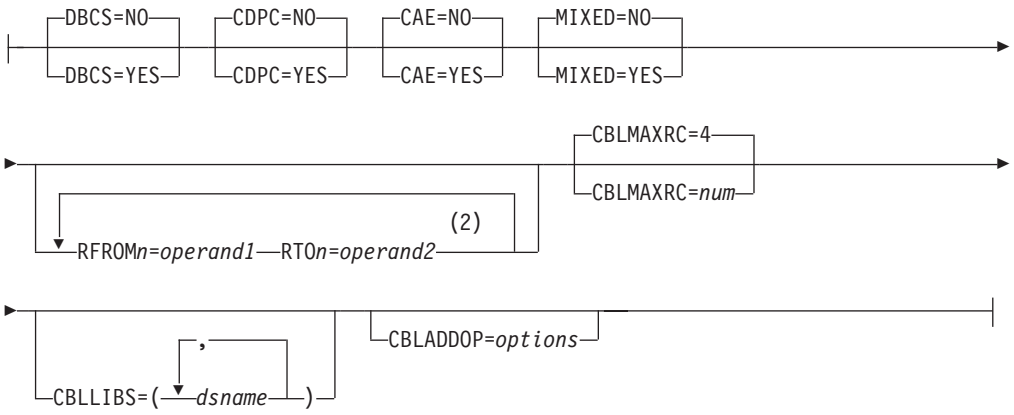
Template options:



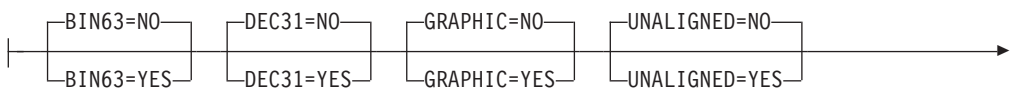
Copybook processing:

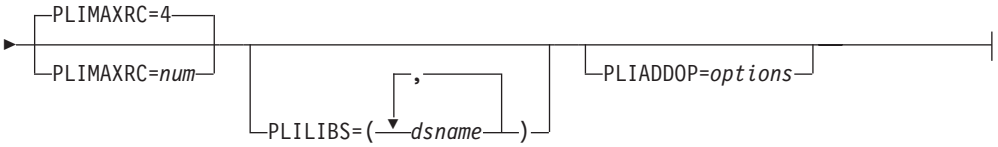


COBOL options:

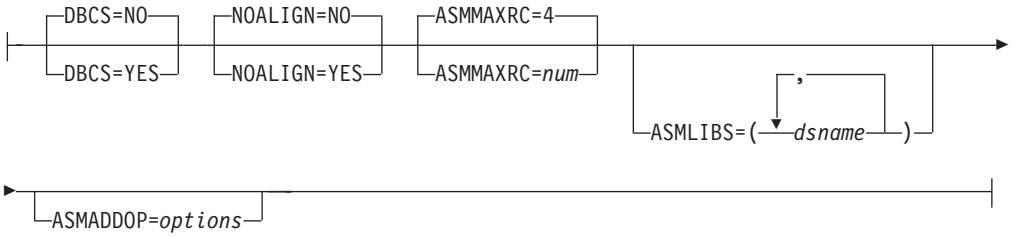


PL/I options:





HLASM options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

DSNIN=*dsname*

Defines the name of the data set to be edited. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. Must be specified when $DSE is invoked from a REXX procedure or TSO clist.

Can be omitted when DSE is entered on a command line. If specified, you bypass the Entry panel, and go directly to the Data Set Edit panel. When omitted, the Data Set Edit Entry panel is displayed in interactive mode and you can specify a valid data set for browsing.

You can further describe this data set, as follows:

MEMBER=*member*

The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the data set is a PDS, you must specify this parameter.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

MEMBER=*member* is ignored if the data set is not a PDS.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

manager
The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuename
The queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

*rt***:***applid***:***rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, browsing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

TCIN=*tcin***(***member***)**
PDS and member name of the copybook or template that describes the record structure of your data set.

OFFSETIN
The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of "From" and "To" pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

For more information on using this function, see:

- "Invoking File Manager panels from outside File Manager" on page 13
- Chapter 12, "Introduction to programming with File Manager functions," on page 397
- "Invoking File Manager panels from REXX procedures or TSO clists" on page 400

```
FILEMGR "$DSE DSNIN='FMNUSER.TEST.KSDS1',KEY=000100"
FILEMGR DSE
```

DSEB (Data Set Edit Batch) — batch only

Purpose
Update disk data set records.

Usage notes
Use this function to update logical records in a single sequential disk data set, a single VSAM data set, one or more members of a PDS, an MQ queue, or a CICS resource. You cannot use DSEB with compressed non-VSAM extended format data sets (compressed PSE data sets).

You can select the records to be processed using:

- Member name selection criteria
- Date created selection criteria
- Date last modified selection criteria
- User ID selection criteria

After the first record in the data set is read, File Manager invokes the REXX procedure specified in the PROC parameter. From now on, the processing of records is controlled by the REXX procedure. This includes the updating of the currently active record, and accessing the next record to be updated. You do not need to move through the data set sequentially. Instead, the REXX procedure can use UP, DOWN, BOTTOM and TOP commands to position to a different record. After you have changed the contents of a record, you must update it before you position to a different record, otherwise the changes are lost.

For procedures requiring REXX, after a record is read, the contents are passed in two File Manager-defined REXX variables, INREC and OUTREC. The contents of the two variables are initially identical. The INREC variable is intended to be used as a reference variable. Any changes made to it are ignored by File Manager. The OUTREC variable can be updated by the exec.

You cannot add records or delete records using DSEB. If you need to add or delete records, you can use one of the File Manager data set copy functions. You cannot change the length of records in a data set using DSEB. If the REXX procedure increases the length of the data in OUTREC or output record (FASTREXX), the data is truncated to its original length before the record is updated. If the REXX procedure decreases the length of the data in OUTREC or output record (FASTREXX), the data is padded to its original length using the pad value specified in the PAD processing option. If no pad value has been specified, the contents of the record are unpredictable.

Performance tips

- DSEB may not be the most efficient function to perform a given task. If an operation is performance critical, first consider using another utility such as DSC, DSP, DSU or FCH.

  DSU is the utility most similar to DSEB, although DSU performs only one pass over the file, from top to bottom.
- DSEB can be run under FASTREXX as long as it adheres to the supported REXX syntax for FASTREXX.
- The DSEB-only File Manager REXX command UP() can be inefficient. If you have a performance-critical DSEB operation which must reprocess records preceding the current record, consider ways of rewriting using the TOP() command, and try to do more work in each forward “pass” over the data.

Options

When you specify the PROC option, you are supplying a REXX procedure. For more information, see the *proc* parameter below.

Return codes

The default return codes from the DSEB function have the following modified meanings:

1 One or more members not updated

2 Change failed (for example invalid key change)

4 No records updated and NOUPDATE=NO

4 No records processed because no members to process

4 No records processed because input empty

4 No records processed because input is in ISPF Packed Data format and the “PACK=STOP” option was specified.

8 REXX non-syntax error encountered while processing records

16 Program Object specified - this is not supported

16 Data set or member in use

16 Data set or member open error

16 Data set not found

16 Other input or output error occurred

16 Insufficient storage available

16 DSEB abended

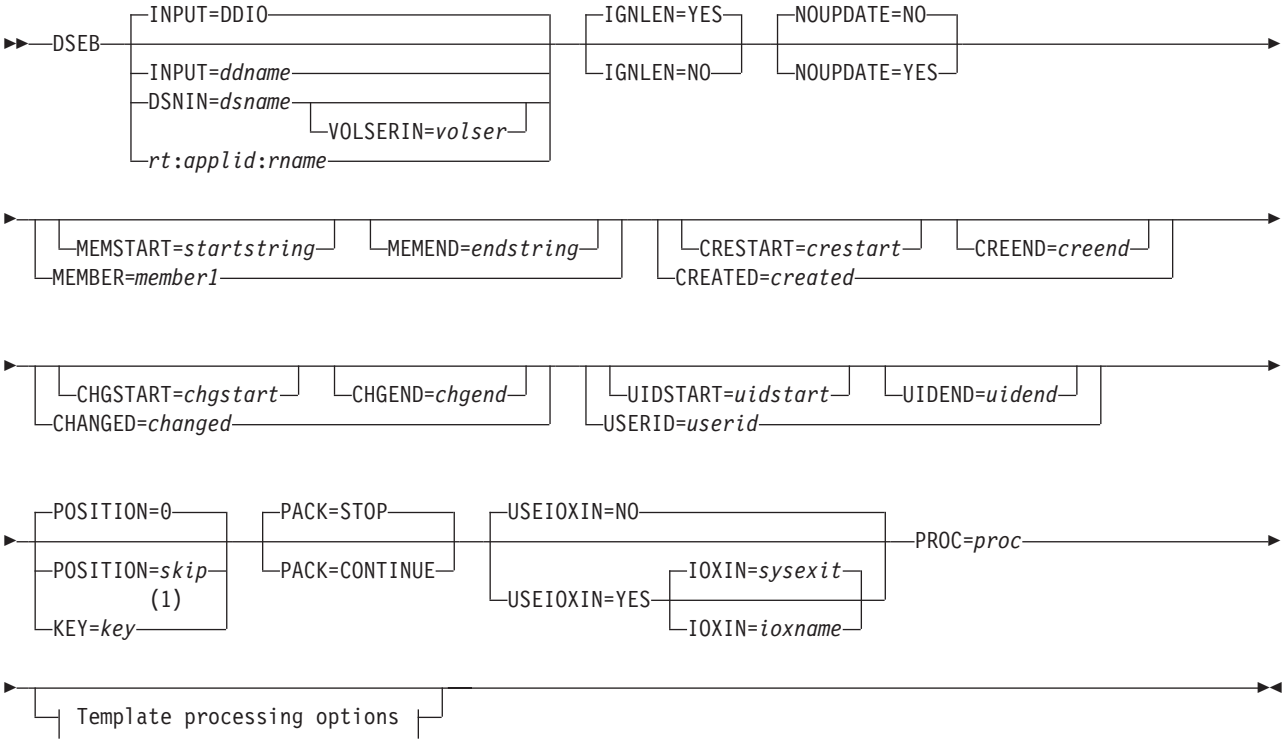
16 Other serious error that stops processing occurred

Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

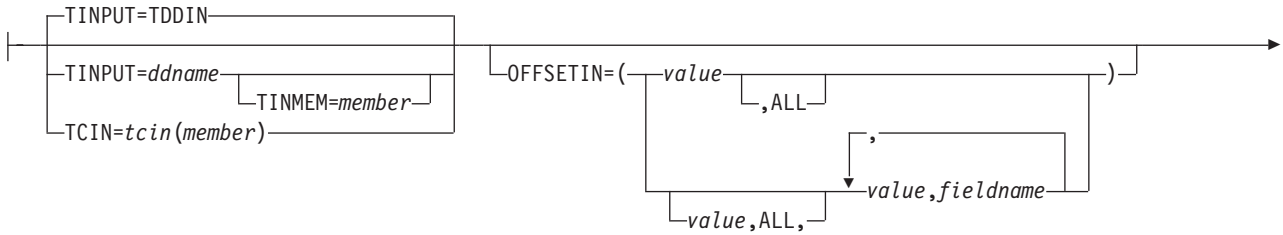
Related functions

DSG Write test data to a data set
DSU Update disk data set records

Syntax

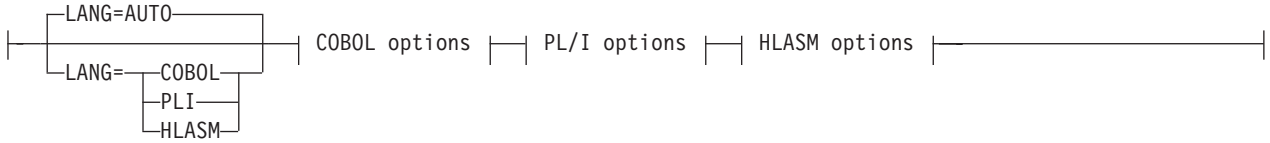


Template processing options:

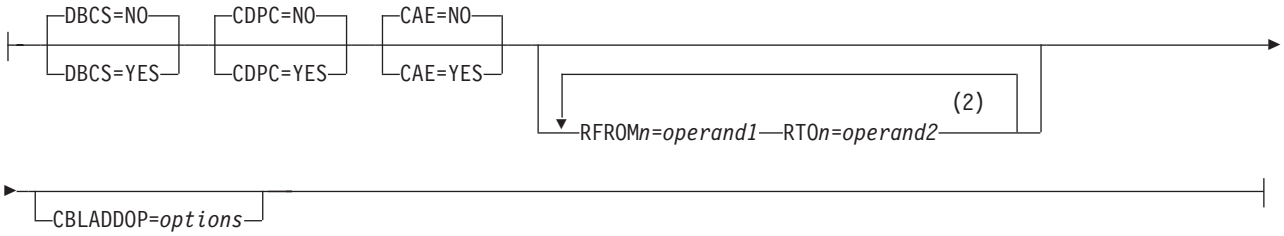


Function reference: DSEB

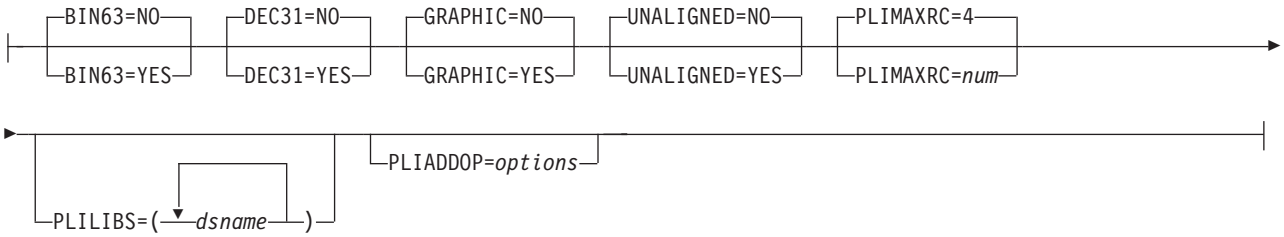
Copybook processing options:



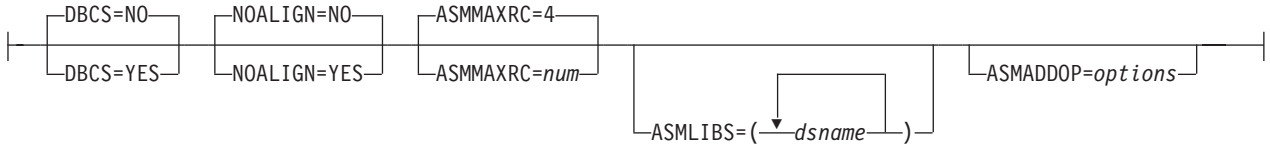
COBOL options:



PL/I options:



HLASM options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

INPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the input data set. The default is DDIN.

DSNIN=*dsname*
Defines the name of the input data set. If specified, any DD statement

provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. You can further describe this data set, as follows:

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

rt:*applid*:*rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

IGNLEN
Specifies whether or not File Manager ignores length mismatches when selecting records for processing.

NO Do not ignore length mismatches. Records that are shorter than the matching structure length in the template are not selected for processing.

YES Use this option to ignore length mismatches.

NOUPDATE
Allows you to specify that you intend no updates to the data set while executing the utility.

NO Updates to the data are honored.

YES Forces the allocation of the data set as input only. UPDATE() function is ignored.

MEMBER=*member1*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the input data set is a PDS(E), you may specify this parameter, or a member name in the DD statement for *ddname*, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains “d” are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member1 is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the copy. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the copy. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the copy.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the copy.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00').

If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record printed. If you omit the *key* and *skip* values, printing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

PACK Determines if File Manager should detect if the input data is in ISPF packed format.

STOP Default. File Manager detects whether the input data is in ISPF packed format, and if it is, stops the processing.

CONTINUE
File Manager does not detect whether the input data is in ISPF packed format and continues processing.

USEIOXIN
Specifies whether to invoke a user I/O exit, to process the input data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the input data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXIN
Specifies the name of the user I/O exit used for the input data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXIN=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXIN has been set to YES and no installation default has been provided, you must specify IOXIN=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

PROC=*proc*
Member name of a REXX procedure that you want to use to process each record before it is updated, or an asterisk (*) to indicate the procedure is inline. If you specify a PDS member name, you must define an FMNEXEC

ddname that identifies the PDS containing the member. If you specify *, the procedure is read from SYSIN immediately following the control statement for the current function. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

For more information about using REXX procedures to process records before they are updated, see Chapter 13, "Enhancing File Manager processing," on page 405.

Template processing

Define which template (if any) is used to describe the record structure in the input data set, and how File Manager processes this template.

TINPUT=*ddname*

Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your input data. The default is TDDIN.

If you specify a concatenated DD, then you must provide the member name, *member*.

TINMEM=*member*

The name of the copybook or template member in the datasets identified by the TINPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCIN parameter is specified.

TCIN=*tcin***(***member***)**

PDS and member name of the copybook or template that describes the record structure of your input data.

Note: If you specify a template for DSEB and DSU, it is ignored, except for calls to the external REXX function PRINT specifying TABL or SNGL format.

OFFSETIN

The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname

The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

Copybook processing

If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG

Determines whether File Manager automatically detect the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO

Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL

Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM

Invoke the HLASM compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the HLASM compilation results in errors.)

COBOL options

The following options are used to compile a COBOL copybook into a template:

DBCS=YES

Use the DBCS compiler option.

DBCS=NO

Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO

Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES

Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO

Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES

Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of "From" and "To" pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

```
//FMNDSEB JOB  'change YPE to ORK every record'
//FMNBAT   EXEC PGM=FILEMGR
//STEPLIB  DD  DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FMNTSPRT DD SYSOUT=*
//SYSIN    DD *
$$FILEM DSEB DSNIN=FMNUSER.KSDS.BASE,
$$FILEM  PROC=*
filerc = 0
do while filerc = 0
  outrec=change(outrec,'YPE','ORK',0)
  update()
  filerc = down(1)
  end
/+
//*
```

This example is the same as Batch example 1, but with FASTREXX.

```
//FMNDSEB JOB  'change YPE to ORK every record'
//FMNBAT   EXEC PGM=FILEMGR
//STEPLIB  DD  DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FMNTSPRT DD SYSOUT=*
//SYSIN    DD *
```

```
$$FILEM DSEB DSNIN=FMNUSER.KSDS.BASE,
$$FILEM  PROC=*
filerc = 0
do while filerc = 0
  outrec=change(outrec,'YPE','ORK',0)
  update()
  filerc = down(1)
  end
/+
//*
//FMNDSEB JOB  '4 recs into 1 rec'
//* Job to take SEQ500.BKUP file, copy it to SEQ500
//* and also create SEQ500.COMBINE, which is created from four input records
//DEL      EXEC PGM=IEFBR14
//COMBINE  DD DSN=FMNUSER.SEQ500.FOURUP,DISP=(MOD,DELETE),
//         SPACE=(TRK,(5,5)),DCB=(LRECL=320,BLKSIZE=0,RECFM=FB)
//FMNBAT   EXEC PGM=FILEMGR
//STEPLIB  DD  DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FMNTSPRT DD SYSOUT=*
//COPY     DD DSN=FMNUSER.SEQ500,DISP=OLD
//COMBINE  DD DSN=FMNUSER.SEQ500.FOURUP,DISP=(NEW,CATLG),
//         SPACE=(TRK,(5,5)),DCB=(LRECL=320,BLKSIZE=0,RECFM=FB)
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM DSEB DSNIN=FMNUSER.SEQ500.BKUP,
$$FILEM  PROC=*
fpos      = 0
recnum    = 0
filerc    = 0
Outrec.combine = ''
do while filerc = 0
  recnum = recnum +1
  /* Create just a copy of this file */
  Outrec.Copy = Outrec
  Write(Copy)
  /* concatenate every four records into one record on combine data set */
  Outrec.combine = Outrec.combine||Outrec
  If recnum // 4 = 0 then do
    Write(Combine)
    Outrec.combine = ''
    end
  filerc = down(1)
  end
/+
//*
```

DSFA (Altering a data set)

Purpose

Alter the attributes of a data set.

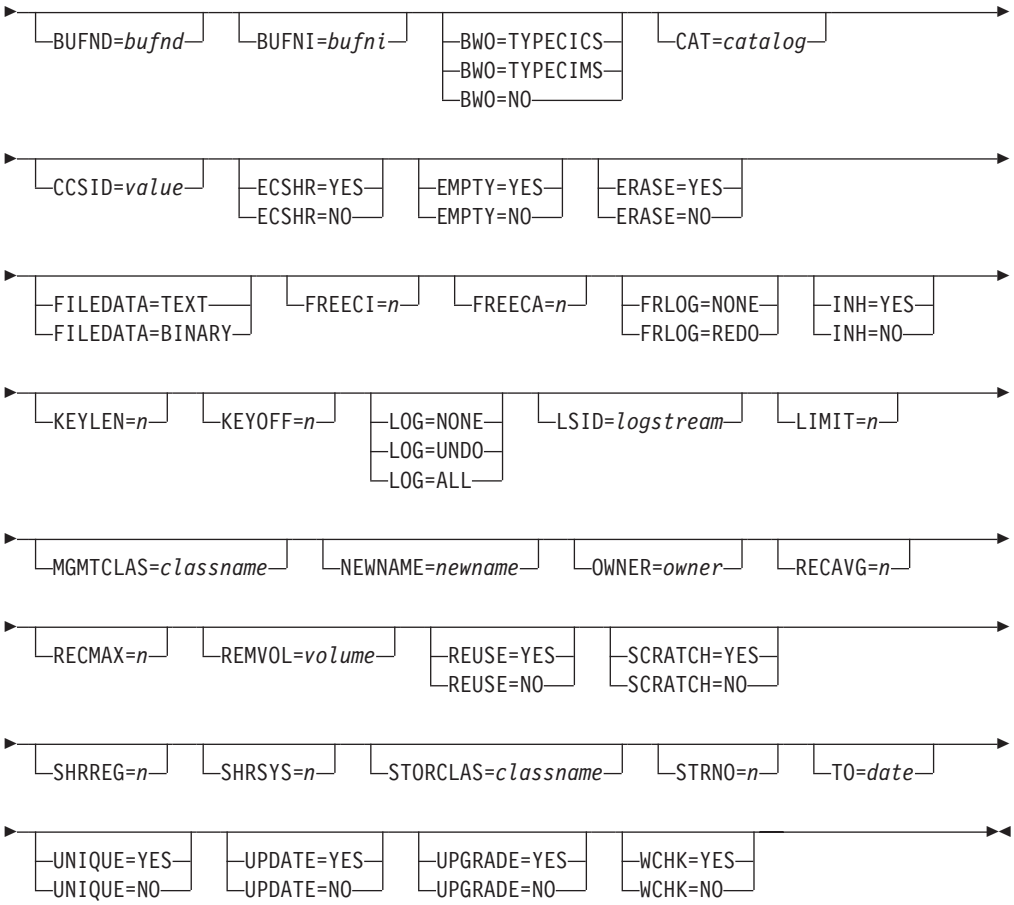
Usage notes

Data set attribute alteration parameters, as supported by Access Method Services, are provided with the exception of: EXCEPTIONEXIT, FILE, LOCK or UNLOCK, and NULLIFY. If the operation cannot be performed, Access Method Services messages are returned; for example, where a parameter is not applicable to the type of data set.

Options

None.

Syntax



DSNIN=*ddname*
The name of the data set whose attributes you want to alter.

Note: All the parameters you specify for the DSFA function are passed to the Access Method Services (AMS) ALTER command.

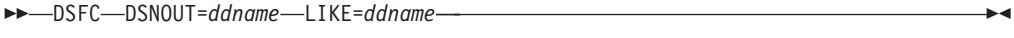
For more information about thes parameters, see *DFSMS AMS for Catalogs*, SC26-7394.

DSFC (Allocate a data set)

Purpose
Allocate a data set based on the attributes of another data set.

Options
None.

Syntax



DSNOUT=*ddname*
The name of the new data set whose attributes are to be based on the attributes of the data set specified for LIKE.

LIKE=*ddname*
The name of the data set whose attributes are to be used when allocating the new data set.

DSFD (Delete a data set)

Purpose
Delete a data set.

Options
None.

Syntax

DSNIN=*ddname*
The name of the data set you want to delete.

DSFR (Rename a data set)

Purpose
Rename a data set.

Options
None.

Syntax

DSNIN=*ddname*
The existing name of the data set that you want to rename.

DSNOUT=*ddname*
The new name of the data set that you want to rename.

DSG (Data Set Generate)

Purpose
Use the DSG function to create and initialize data values in VSAM data sets, sequential data sets, PDS members, HFS files, MQ queues, or CICS resources.

You specify the number of records that are to be created and how they are to be initialized. You can initialize the data with fill characters and patterns. When you use a copybook or a template you can initialize records at the field level. You can change the field create attributes for individual fields by editing the copybook or template.

Options
You can specify a fill character or fill pattern, a sequence field, and a block size.

Return codes
The default return codes from the DSG function have the following modified meanings:

4 No records generated because zero records specified

16 No records generated because DISP=MOD specified for member

16 No records generated because key not fully within data

16 Member name required and not specified

16 Program Object specified - this is not supported

16 Data set in use

16 Data set or member in use

16 Data set or member open error

16 Other input or output error occurred

16 Insufficient storage available

16 DSG abended

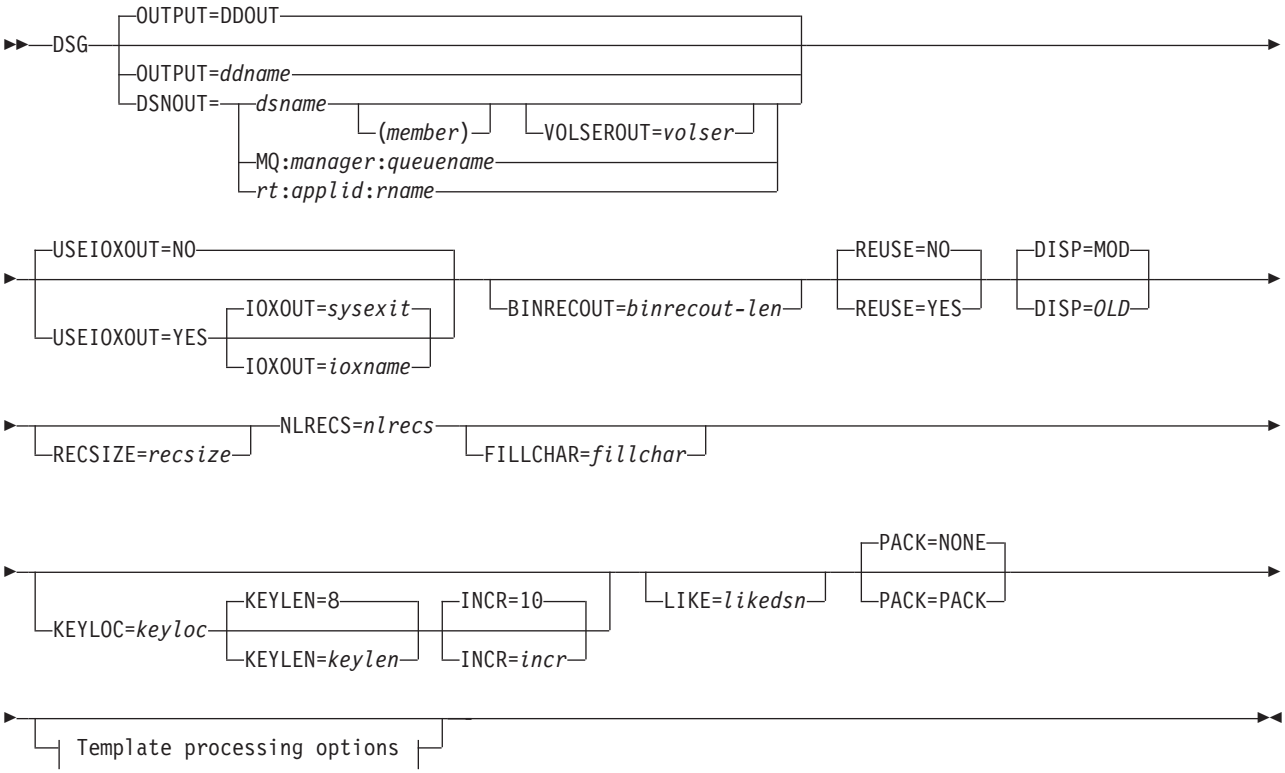
16 Other serious error that stops processing occurred

Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Related functions

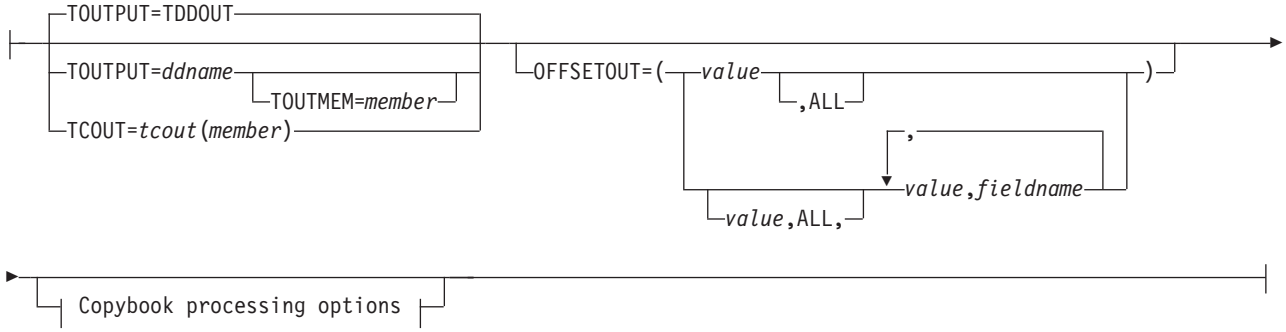
BT Write test data to a tape file
INT Initialize a tape

Syntax

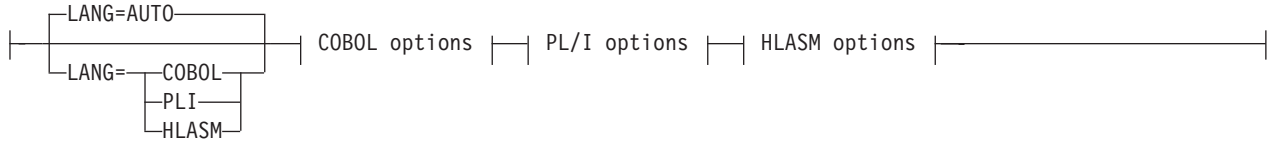


Function reference: DSG

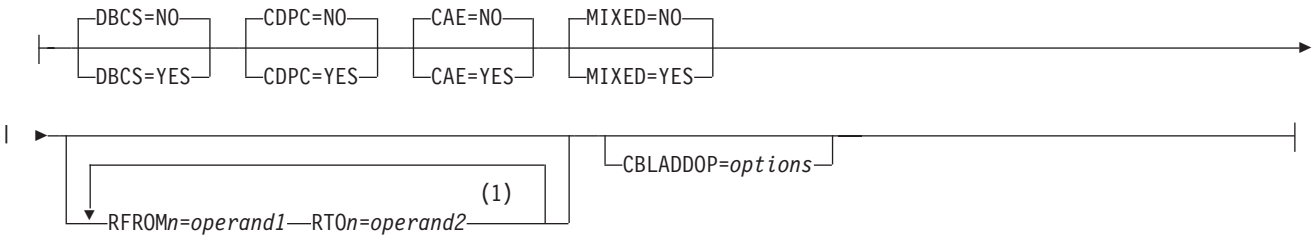
Template processing options:



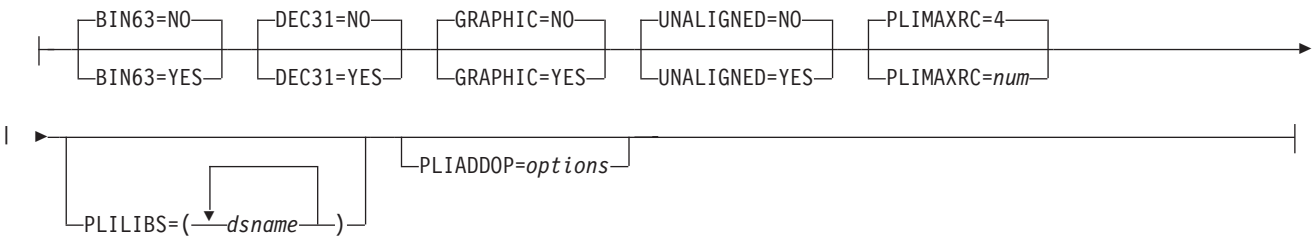
Copybook processing options:



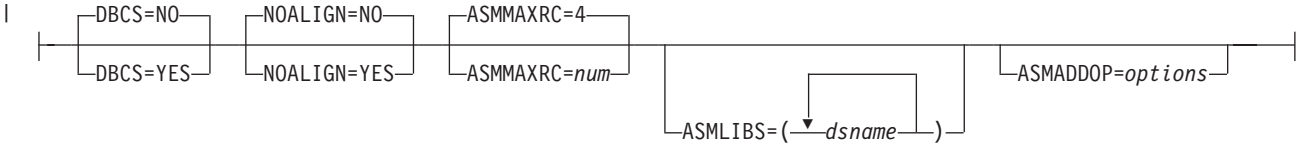
COBOL options:



PL/I options:



HLASM options:



Notes:

1 *n*= 1 to 5.

OUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the output data set or HFS file. The default is `DDOUT`.

DSNOUT=*dsname*
Defines the name of the output data set or an absolute path to the output HFS file. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it overmore than one line. You can further describe this data set, as follows:

(*member***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

VOLSEROUT=*volser*
Volume serial number for a new or non-cataloged data set.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

manager
The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuename
The queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

*rt***:***applid***:***rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

USEIOXOUT
Specifies whether to invoke a user I/O exit, to process the output data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the output data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXOUT
Specifies the name of the user I/O exit used for the output data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXOUT=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXOUT has been set to YES and no installation default has been provided, you must specify IOXOUT=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

BINRECOUT=*binrecout-len*
Specifies the record length for processing the HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

REUSE
Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

DISP Disposition of the output data set. Specify OLD or MOD.

OLD Writes input records to the existing output data set, starting from the beginning.

MOD Default. Appends the input records to the end of the existing output data set.

Note: If you are appending to a sequential data set, ensure that the existing data set does not contain data in ISPF Packed Data format. Appending records to a packed data set results in data errors.

Note: SMS might modify the allocation of new data sets on your system. For details, contact your SMS Administrator.

RECSIZE=*recsize*
1-9999999, depending on the output file characteristics.

For fixed record formats the record length is set to the record length and user input is ignored.

For undefined record formats the record length is set to the block length and user input is ignored.

For variable record formats the record length specified is used if it is less than the maximum record size. Otherwise the maximum record size is used and user input is ignored.

When the data create function is used with a template, the length of the records written depends upon whether the records are fixed-length or variable-length, and the length of the record description in the template (note that the length of the record in the template must always be less than or equal to the data set record length). The value specified in *recsize* is ignored when using a template.

For fixed-length records created with a template, all records have the length specified for the data set. If the length of the record in the template is less than the data set record length, the records are padded with the fill character.

For variable-length records created with a template, the length of each record depends on the length of the record built using the field attributes in the template. If the template record description contains one or more variable-length arrays, the length of the record varies according to the value or values assigned to objects of the OCCURS DEPENDING ON clauses.

NLRECS=*nlrecs*
Number of logical records to be written. The maximum is 999 999 999.

FILLCHAR=*fillchar*
To fill each byte of each record with data, specify one of the following:

char A character, such as 0 to be written in each byte. (To specify a lowercase character such as 'a', use either C'a' or X'81'.)
X'*cc***'** A binary character, such as X'04' to be written in each byte.
AN To write alphanumeric characters (A to Z and 0 to 9).
BIN To write binary characters (X'00' to X'FF').
RAND
To write random binary characters (X'00' to X'FF').

If you specify AN or BIN, characters are written in a “ripple” pattern. For example, if you specify AN, the first 3 records look like this:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789AB
CDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABC
```

The default is a blank.

KEYLOC=*keyloc*
Position (starting with 1) of the sequence field within the output records. If omitted, there is no sequence field.

Note: KEYLOC is not the same as a VSAM KSDS key position. KEYLOC refers to a "virtual" key field (sequence number) created in all records to identify a specific record within a file.

KEYLEN=*keylen*
Length of the sequence field, from 1 to 9. The default is 8.

Note: KEYLEN is not the same as a VSAM KSDS key length. KEYLEN refers to a "virtual" key field (sequence number) created in all records to identify a specific record within a file.

INCR=*incr*
Increment value for the sequence field. The default is 10.

LIKE=*likedsname*
Name of the data set to be used as the model for the output allocation.

File Manager uses the TSO prefix (as defined for your user ID) used as the high-level qualifier for any name that is not enclosed in quotes.

If copying data sets, “=” denotes the current input data set.

The model data set must reside on a DASD volume, be cataloged, and SMS must be active.

PACK Determines if File Manager writes the output data in ISPF packed format. This keyword is ignored when processing VSAM data sets. When an I/O exit has been specified for either the input or output data set (or both), the only valid option is PACK=NONE.

PACK Instructs File Manager to write the output in ISPF packed format.

NONE
Instructs File Manager not to write the output in ISPF packed format.

Note: Do not use the DISP=MOD and PACK=PACK option at the same time. Appending packed data to a packed or non-packed sequential data set results in data errors. Appending non-packed data to a packed data set also results in data errors.

Template processing
The template processing parameters define which template is used to describe the structure of the new records and how File Manager processes the template.

TOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your output data. The default is TDDOUT.

If you specify a concatenated DD, then you must provide the member name, *member*.

TOUTMEM=*member*
The name of the copybook or template member in the datasets identified by the TOUTPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCOUT parameter is specified.

TCOUT=*tcout***(***member***)**
PDS and member name of the copybook or template that describes the record structure of your output data.

OFFSETOUT
The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the "To" template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the "To" template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is applied. The default is the first Level 01 field in the "To" template.

Copybook processing options

If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG

Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO

Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL

Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM

Invoke the HLASM compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the HLASM compilation results in errors.)

COBOL options

The following options are used to compile a COBOL copybook into a template:

DBCS=YES

Use the DBCS compiler option.

DBCS=NO

Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO

Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES

Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO

Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES

Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of "From" and "To" pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'=====' ` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

```
//DSG  JOB (acct),'name'  Create VSAM Data
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//IDCPROC  PROC
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//*  EXAMPLE DSG JOB
//*  17 RECORDS ARE WRITTEN TO A NONINDEXED
//*  VSAM FILE.  THE OUTPUT FILE CONTAINS
//*  ALPHANUMERIC CHARACTERS.
//*
//DMBBV02   EXEC IDCPROC,REGION=6M
//SYSIN    DD *
 DELETE FMNUSER.VSAM.DSG25 CLUSTER PURGE ERASE
 DEFINE CLUSTER ( -
      NAME( FMNUSER.VSAM.DSG25 ) -
      CISZ(4096) -
      FREESPACE(30 30) -
      NONINDEXED -
      KEYS(12 0) -
      RECORDS(1000 100) -
      RECORDSIZE(234 234) -
      REUSE) -
      DATA( NAME(FMNUSER.VSAM.DSG25.DATA) )
/*
//*
//STP1005     EXEC FMBAT
//DDOUT  DD DISP=SHR,DSN=FMNUSER.VSAM.DSG25
//SYSIN    DD *
$$FILEM VER
$$FILEM DSG OUTPUT=DDOUT,REUSE=NO,NLRECS=17,
$$FILEM     RECSIZE=234,FILLCHAR=AN
$$FILEM EOJ
/*
```

This example shows how to define a new data set set using a "model" data set (supports VSAM and non-VSAM data sets) and fill it with some data.

Note that the new data set is allocated based on the "model" data set as described by the LIKE parameter.

```
 //STP1006 EXEC FMBAT
$$FILEM DSG DSNOUT=TEST.BASE.NEW,
$$FILEM LIKE='FMN.TEST.BASE.RFM0041.SEGMENT.PS',
$$FILEM DISP=MOD,
$$FILEM PACK=NONE,
$$FILEM NLRECS=100
/*
```

DSI (Data Set Information)

Purpose

Display detailed dataset information.

Usage notes

Use this function to print the details about a data set. The data set information printed (returned) corresponds to the INFO primary command output for online processing.

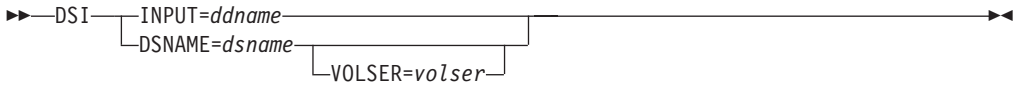
Options

None.

Related functions

DSX Print disk dataset EXTENT information
DVT Print VTOC entries

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

dsname

Name of a disk data set.

volser Volume serial number for a non-cataloged data set.

```
//DP JOB (acct),'name' Data Set Info
//*
//FMBAT PROC
//FMBAT EXEC PGM=FILEMGR
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=* /
/ PEND
//*
//FILEMGR EXEC FMBAT
//DISK DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN DD *
$$FILEM DSI INPUT=DISK
$$FILEM DSI DSN=SYS1.PROCLIB equivalent DSN use
$$FILEM EOJ
/*
```

DSM (Data Set Compare)

Purpose

Use the DSM function to:

- Compare data from any partitioned, sequential or VSAM data set, HFS file, MQ queue, or CICS resource to data in any other partitioned, sequential or VSAM data set, HFS file, MQ queue, or CICS resource.
- Perform a field level comparison. By using an “Old” copybook or template with a “New” copybook or template, you can compare selected fields with the result of the comparison reflecting the types of data in the fields.
- Perform a load module comparison. Load module and CSECT information from both the "Old" and "New" versions of the module is extracted and compared. By specifying compare criteria, you can see differences between specific properties of the load modules, such as load module size, link date, CSECT names, and compilers used.
- Produce a comparison report, showing information such as where insertions, deletions or changes have occurred in the “New” data set. The report's content and structure reflects the various comparison options used.
- Create output data sets containing records identified as inserted, deleted, old and new changed records, and old and new matched records.

Usage Notes

- Select the records to be compared, using:
  - The start key (VSAM only)
  - The skip field
  - The compare count field
  - Conditional expressions defined in the “Old” and/or “New” templates.
  - The "Number of differences to report" option

  To perform a field level comparison, you must provide an “Old” and a “New” copybook or template and use the TYPE=FORMATTED comparison option. You can use the field mapping specified in a “New” template (created online), or you can use the default mapping generated from the template or copybook contents, or you can specify the field mapping in the batch file, using the FIELDOLD and FIELDNEW keywords.
- Specify the way in which the comparison is performed, using:
  - The compare options
  - The synchronization options

  If SYNCH=KEYED is used, up to sixteen key segments can be specified to create a single composite key.
- Specify the type of output produced and the way in which the output is displayed, using:
  - The listing type
  - The listing options
- To specify a field name containing lowercase or mixedcase characters, use the character literal form of the name surrounded by quotes and preceded with character C. For example, C'aBc'.

Performance tips

- DSM was designed with a focus on comparing data in fields using templates or copybooks. See the template performance tips in "General tips about performance when you use File Manager functions" on page 833.
- The ISPF utility SuperC may perform more efficiently when comparing ordinary text data sets, since the special features of File Manager are not required. For more details on SuperC, see the *z/OS ISPF User's Guide Vol II*.

Return codes

The default return codes from the DSM function have the following modified meanings:

0 The function was completed successfully and the compare sets match.

1 The function was completed successfully and the compare sets do not match.

2 One of the compare sets was empty, so no comparison was performed.

4 Both of the compare sets were empty, so no comparison was performed.

4 No comparison was performed because one of the input data sets or members in ISPF Packed Data format and the "PACK=SKIP" option was specified.

4 At least one record with an unmapped type was encountered.

8 A data error occurred, for example, a key sequence error was found when using a keyed comparison. The conditions that result in a return code of 8 are:

For keyed synchronization:

key truncation error
A key truncation error occurs when a key segment falls outside the record.

key sequence error
A key sequence error occurs when the key for a record is found to be less than or equal to the key for the previous record.

For read-ahead synchronization:

read-ahead resynchronization failure
Read-ahead resynchronization fails when matching records cannot be found within the read-ahead limit.

16 No records compared because input and output physically the same.

16 Invalid data in template.

16 Data set or member in use.

16 Data set or member open error.

16 Data set or member not found.

16 Other input or output error occurred.

16 Member name required and not specified.

16 Insufficient storage available.

16 DSM abended

16 Input data appears ISPF packed but is not valid.

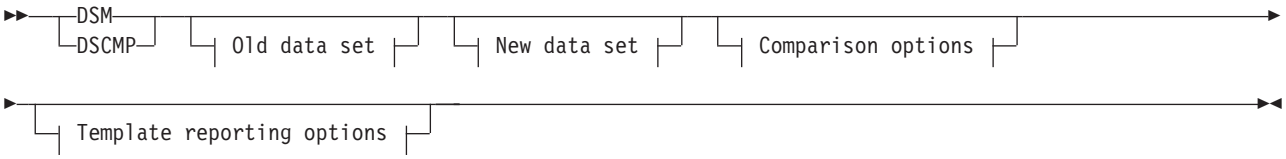
16 Other serious error that stops processing occurred.

16 A severe error occurred, causing File Manager to terminate.

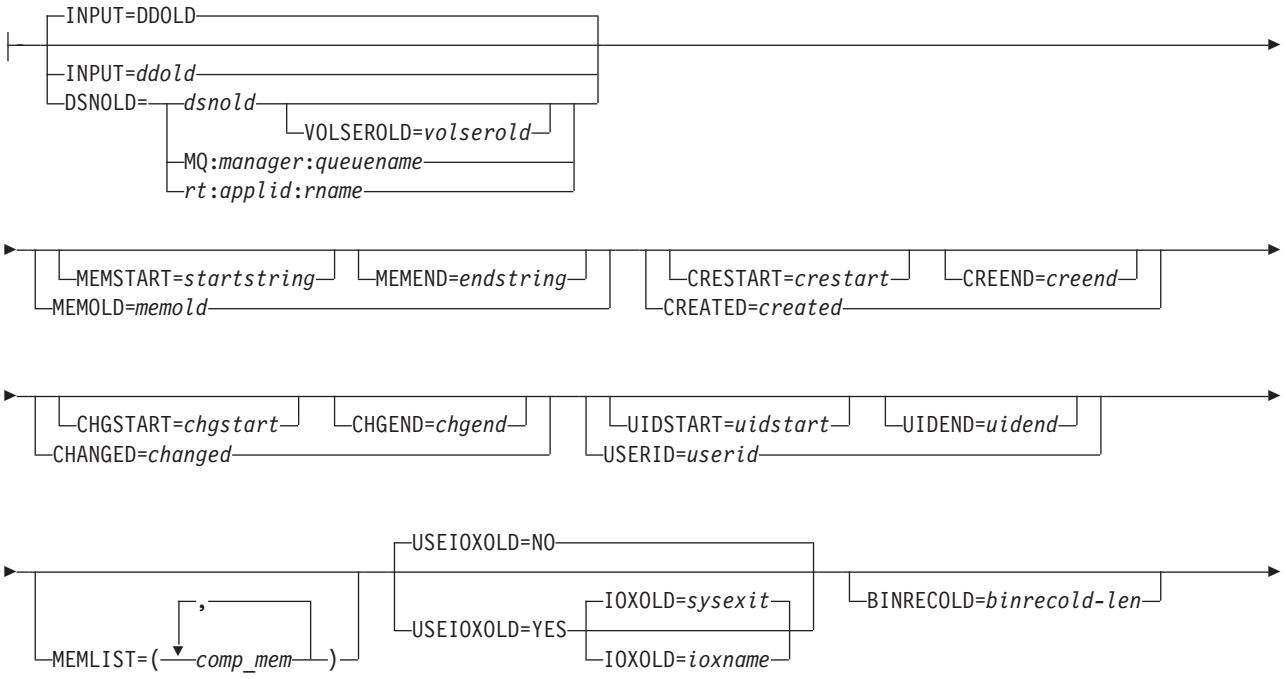
Tip: When handling return codes 0, 1, 2, and 4 in your JCL, you might choose to deal with each code separately or you might choose to deal with return codes 0 and 4 as a single result (the compare sets match) and 1 and 2 as another result (the compare sets do not match).

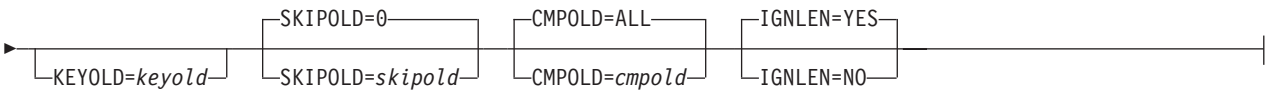
Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Syntax

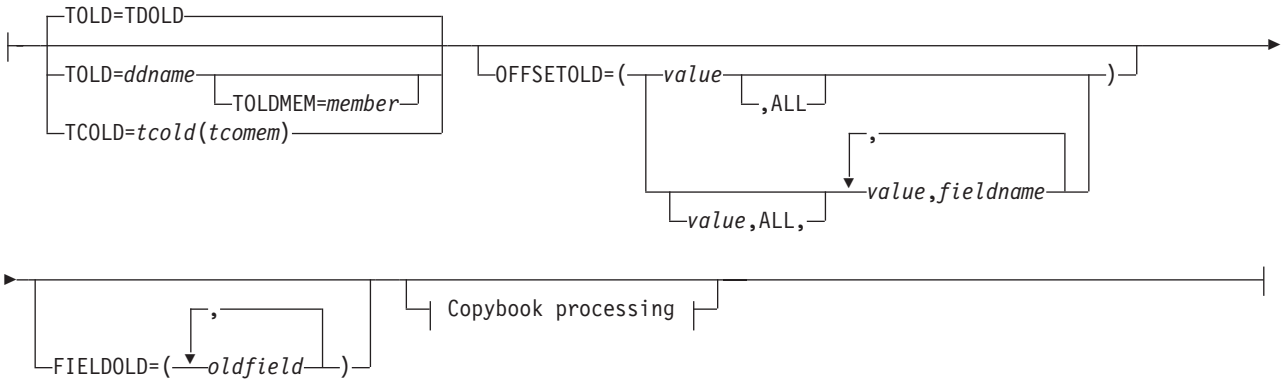


Old data set:

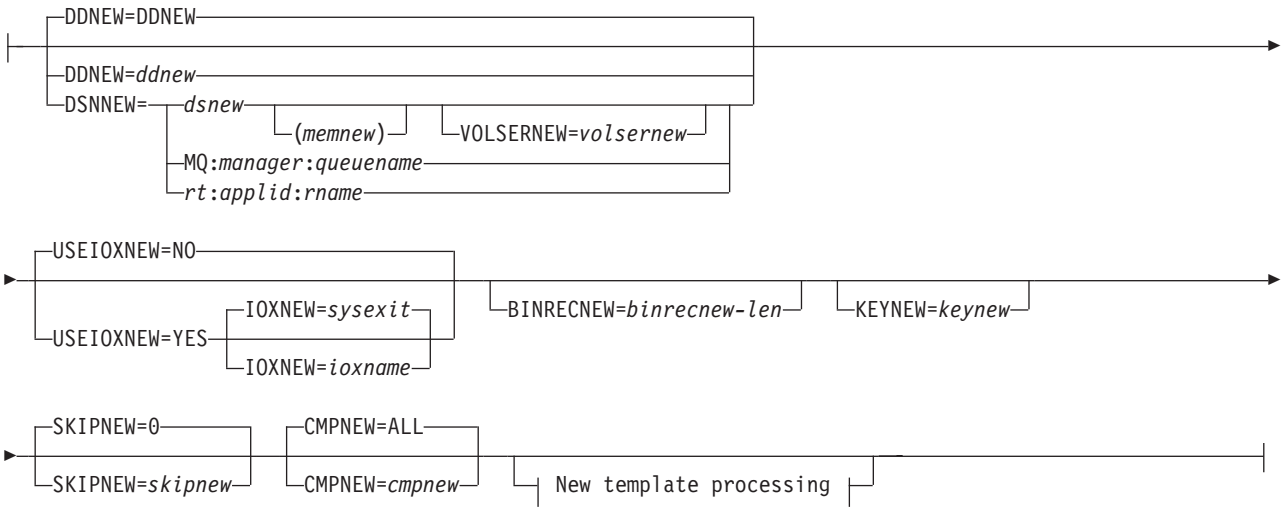




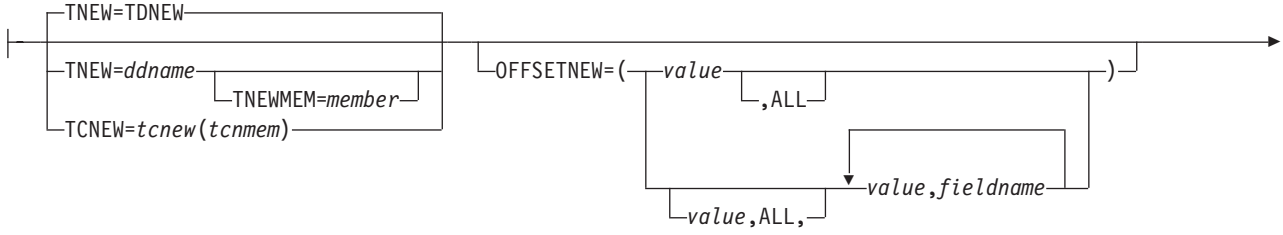
Old template processing:



New data set:

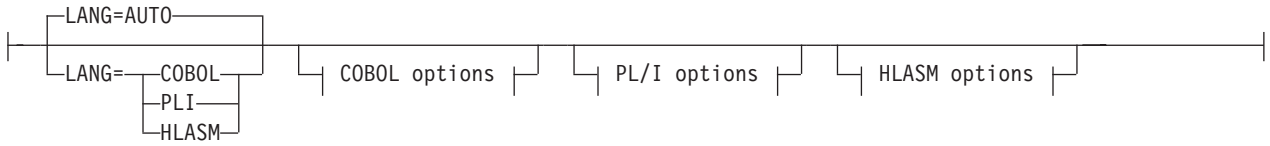


New template processing:

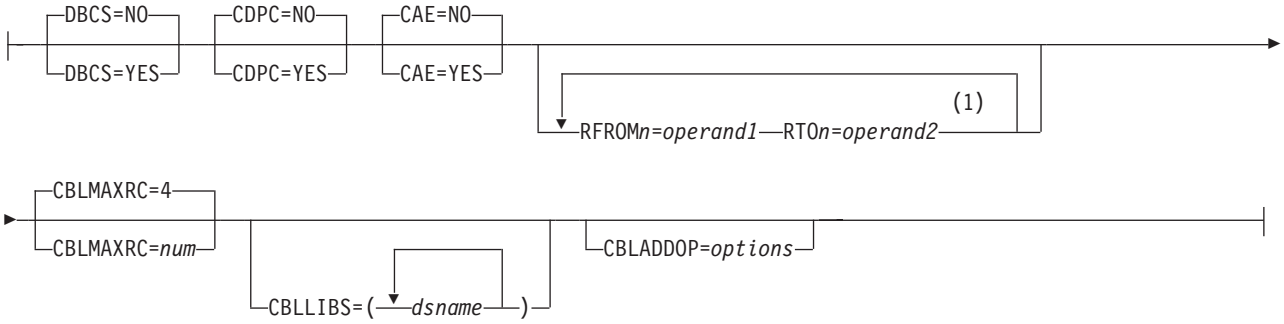


Function reference: DSM

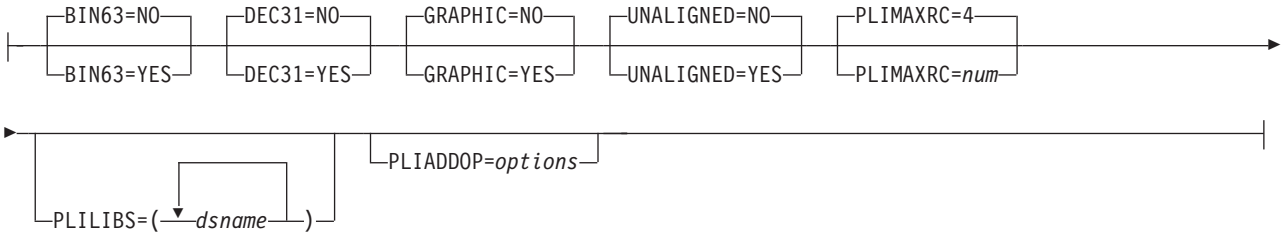
Copybook processing:



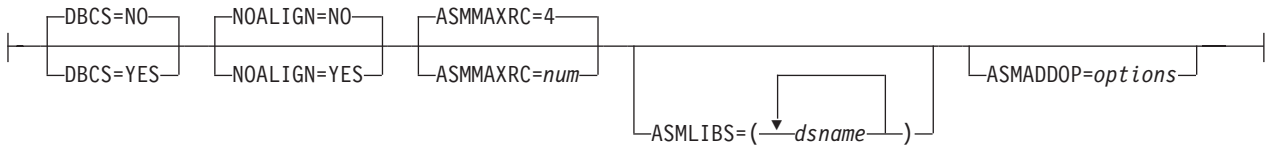
COBOL options:



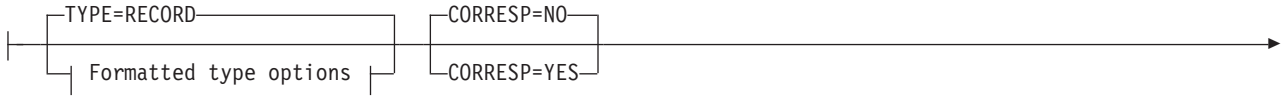
PL/I options:

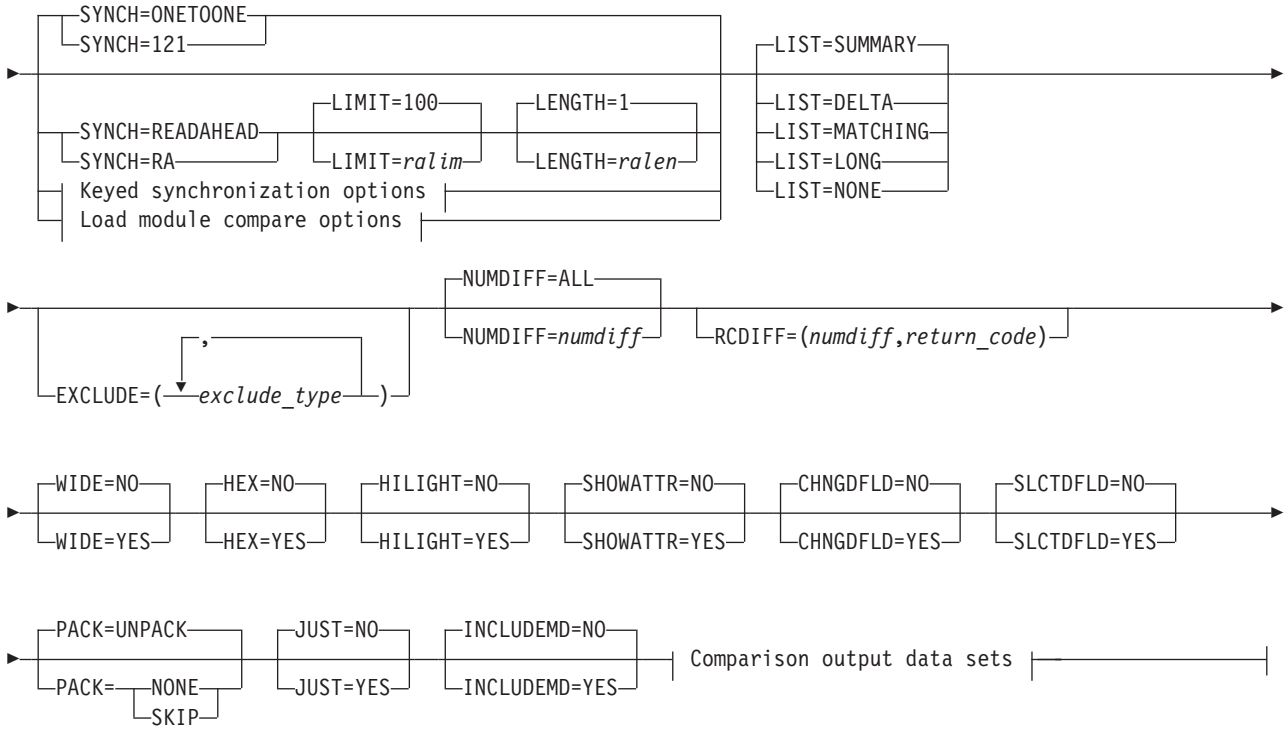


HLASM options:

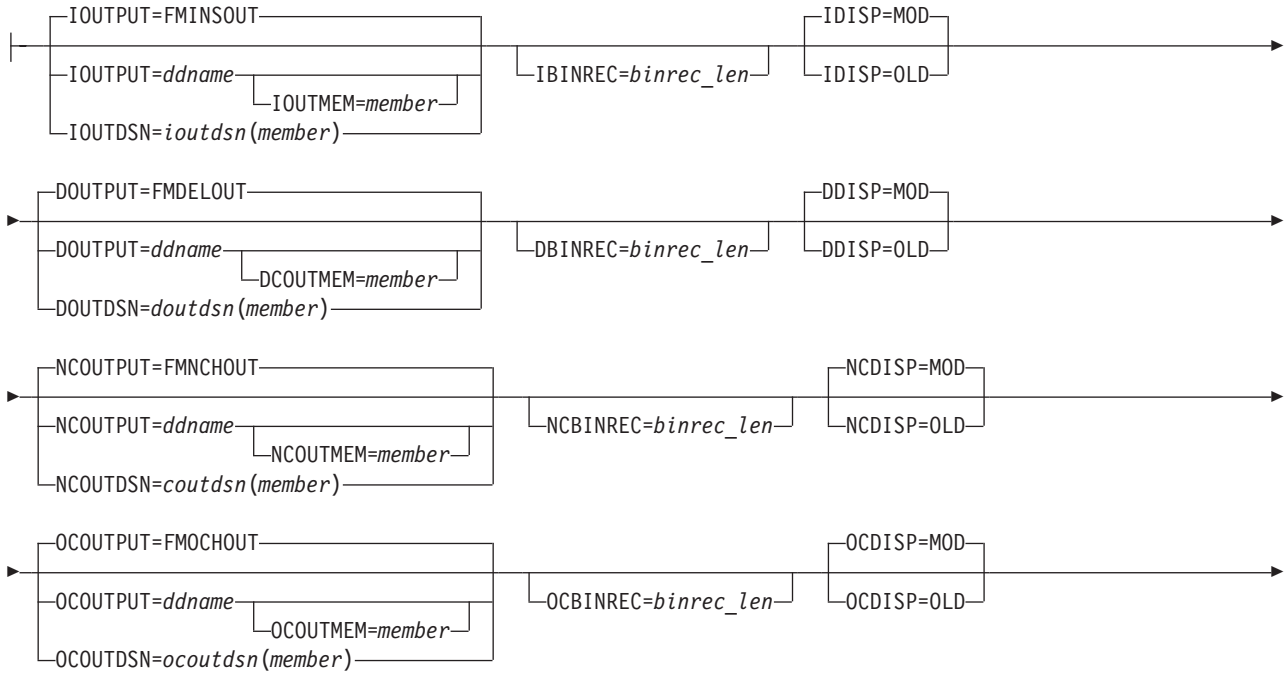


Comparison options:

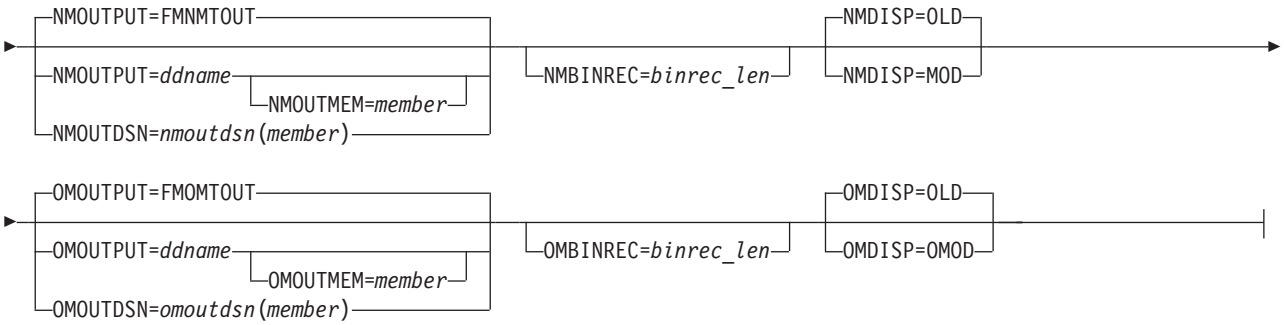




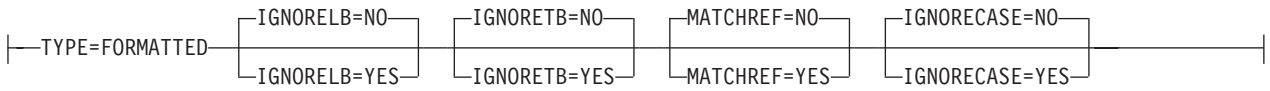
Comparison output data sets:



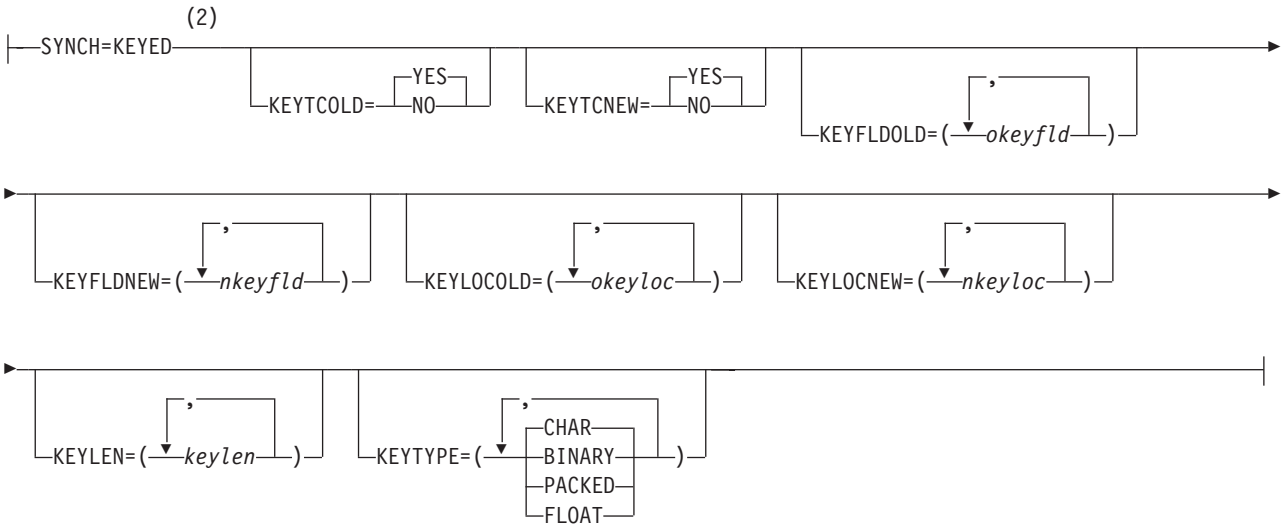
Function reference: DSM



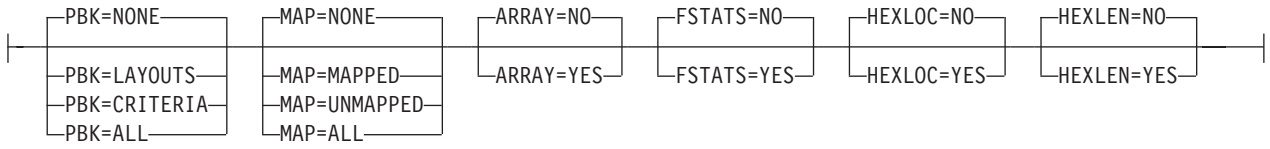
Formatted type options:



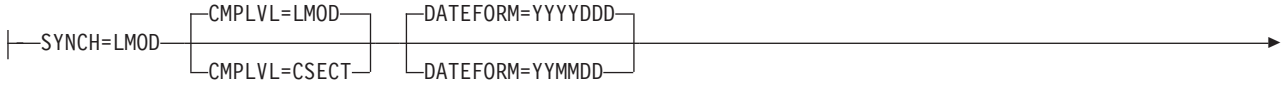
Keyed synchronization options:

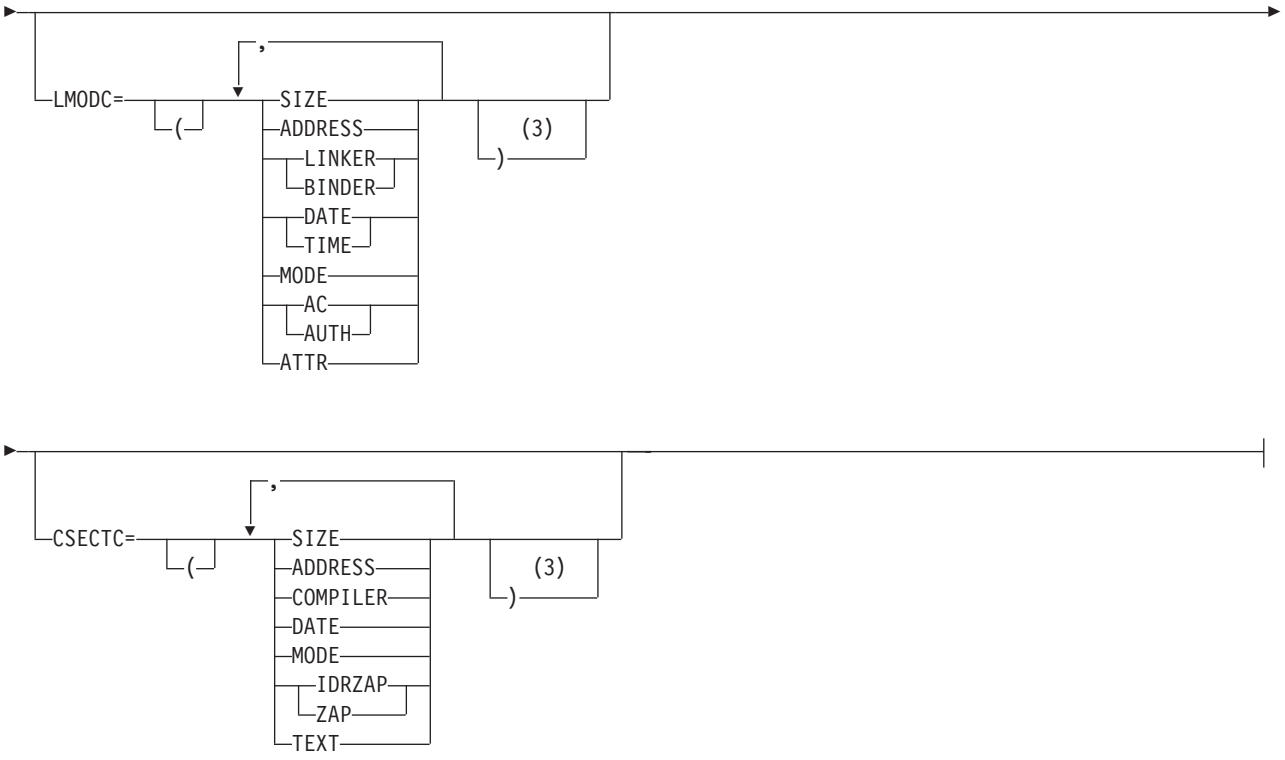


Template reporting options:



Load module compare options:





Notes:

1 *n*= 1 to 5.

2 All of the following keywords are shown as optional in the syntax, but in practice the actual requirements are dependent on a number of factors, as described in the SYNCH=KEYED definition below the syntax diagram.

3 Provide closing bracket when opening bracket has been used.

Old data set specifications

The "Old" data set can be specified as follows:

DDOLD=*ddold*

Defines a reference to a DD or TSO ALLOC statement for the "Old" data set or HFS file. The default is DDOLD.

DSNOLD=*dsnold*

Defines the name of the "Old" data set or an absolute path to the "Old" HFS file. If specified, any DD statements provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated MEMOLD parameter must be empty. An absolute path to an HFS file must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line. To further describe the data set, use the following:

VOLSEROLD=*volserold*

The VOLUME serial number for a non-cataloged "Old" data set.

MEMOLD=*memold*

The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS library. You can

specify this parameter, or a member name in the DD statement for *ddname*, or specify a member or members in the MEMLIST parameter, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name, and two special pattern characters: the asterisk (*) and the percent symbol (%).

An * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of *d*, all members in the PDS whose name contains "d" are processed.

A % is a place-holding character that represents a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the PDS with a 4-character name are processed.

MEMOLD is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the compare. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *memold* parameter of the MEMOLD keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the compare. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *memold* parameter of the MEMOLD keyword.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

managerold
The “Old” MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuenameold
The “Old” queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

*rt***:***applid***:***rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the "Old" data set is a PDS(E), you can specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates, or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the compare.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the compare.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the "Old" data set is a PDS(E), you can specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates, or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the compare.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the compare.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the "Old" data set is a PDS(E), you can specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the compare.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the compare.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00'). If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

MEMLIST
Provides a means of selecting members from a PDS where no generic name pattern and no member name range has been specified. If the MEMLIST keyword is specified, only those

members included in the MEMLIST arguments are compared with the corresponding members in the output data set. Members selected by the MEMBER=*memold* that are not included in the MEMLIST arguments are not compared.

comp_mem
The name of the member to be compared. Generic name masks are not allowed.

USEIOXOLD
Specifies whether to invoke a user I/O exit, to process the "Old" data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the "Old" data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXOLD
Specifies the name of the user I/O exit used for the "Old" data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXOLD=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXOLD has been set to YES and no installation default has been provided, you must specify IOXOLD=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

BINRECOLD=*binrecold-len*
Specifies the record length used for processing the "Old" HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

KEYOLD=*keyold*
A key for KSDS records or a slot number for RRDS records, for the "Old" data set. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record compared. If you omit the *keyold* and *skipold* values, the comparison begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

SKIPOLD=*skipold*
Number of logical records to be skipped from the beginning of the "Old" data set. The default is 0.

CMPOLD=cmpold
Number of records from the "Old" data set to be compared. The maximum number is 999 999 999. If you specify ALL or omit the parameter, all the remaining records are compared.

IGNLEN
Specifies whether or not File Manager ignores length mismatches when selecting records for processing.

NO Do not ignore length mismatches. Records that are shorter than the matching structure length in the template are not selected for processing.

YES Use this option to ignore length mismatches.

Old template processing
Use these options to specify the "Old" copybook or template that describes the record structure of your "Old" data set.

TOLD=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your "Old" data set. The default is TDOLD.

If you specify a concatenated DD, then you must provide the member name, *member*.

TOLDMEM=*member*
The name of the copybook or template member in the datasets identified by the TOLD parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCOLD parameter is specified.

TCOLD=*tcold***(***tcomem***)**
PDS and member name of the "Old" copybook or template that describes the record structure of your "Old" data set.

OFFSETOLD
The length of the 01 field in the "Old" template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the "Old" template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the "Old" template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the "Old" template.

FIELDOLD=(*oldfield1***,***oldfield2***,...)**
Field name or names in the "Old" template, used to create a mapping for a formatted comparison. This keyword is used in conjunction with the FIELDNEW keyword to define mapping within the batch file. The parentheses are optional when only one field is specified but mandatory when more than one field is included. The names are mapped in the order given, that is, *oldfield1* is mapped to *newfield1* and so on, overriding any default or existing mapping.

FIELDOLD and FIELDNEW can be used with or without an existing mapping in the template and with or without the CORRESP keyword setting, as follows:

Table 14. Batch mapping behavior

Specifications			
"New" template contains mapping?	**CORRESP=?**	**FIELDxxx specified?**	**Behavior**
no	NO	no	map corresponding fields
		yes	map specified fields
	YES	no	map corresponding fields
		yes	map corresponding fields then remap specified fields
yes	NO	no	use mapping in TCNEW template
		yes	use mapping in TCNEW template then remap specified fields
	YES	no	map corresponding fields
		yes	map corresponding fields then remap specified fields

Note: "Specified fields" refers to those fields specified in the FIELDOLD and FIELDNEW arguments.

Handling multiple 01s and duplicate field names

Field mapping specifications that are created via the FIELDOLD and FIELDNEW keywords are applied on a first match basis. For example, if the "Old" template came from a copybook containing:

```
01 OLD-TYPE01.
   03 BINARY-X              PIC 999999999 USAGE BINARY.
01 OLD-TYPE02.
   03 BINARY-1              PIC 999999999 USAGE BINARY.
01 OLD-TYPE03.
   03 BINARY-1              PIC 999999999 USAGE BINARY.
```

and the "New" template came from a copybook containing:

```
01 NEW-TYPE01.
   03 BINARY-2              PIC 999999999 USAGE BINARY.
01 NEW-TYPE02.
   03 BINARY-2              PIC 999999999 USAGE BINARY.
```

```
   05 DUP-FIELD.
   07 BINARY-2              PIC 999999999 USAGE BINARY.
01 NEW-TYPE03.
   03 BINARY-2              PIC 999999999 USAGE BINARY.
```

a mapping specification of:

```
$$FILEM FIELDOLD=BINARY-1,
$$FILEM FIELDNEW=BINARY-2,
```

maps the 03 BINARY-2 field in NEW-TYPE02 to the BINARY-1 field in OLD-TYPE02. This is because the 03 BINARY-2 field in NEW-TYPE02 is the first "New" template field found named BINARY-2, where the corresponding 01-level, OLD-TYPE02, contains a field called BINARY-1. The 03 BINARY-2 field in NEW-TYPE01 is not mapped because the corresponding 01-level, OLD-TYPE01, does not contain a 03 BINARY-1 field.

You can override this default behavior to specify different field mappings by using a dot qualification. For example,

```
$$FILEM FIELDOLD=BINARY-1,
$$FILEM FIELDNEW=DUPFIELD.BINARY-2,
```

would map the 07 BINARY-2 field in NEW-TYPE02 to the BINARY-1 field in OLD-TYPE02.

```
$$FILEM FIELDOLD=BINARY-1,
$$FILEM FIELDNEW=NEW-TYPE03.BINARY-2,
```

would map the 03 BINARY-2 field in NEW-TYPE03 to the BINARY-1 field in OLD-TYPE03.

Note:

1. This only affects the mapping. Normal record identification procedures must be understood and employed to ensure the correct 01 is in effect for each record comparison.
2. The qualifiers are resolved from left to right, skipping over levels not present in the qualification so that only enough information to uniquely identify a field need be provided.

New data set specifications

The "New" data set can be specified as follows:

DDNEW=*ddnew*

Defines a reference to a DD or TSO ALLOC statement for the "New" data set or HFS file. The default is DDNEW.

DSNNEW=*dsnnew*

Defines the name of the "New" data set or an absolute path to the "New" HFS file. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated MEMNEW parameter must be empty. An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line. To further describe the data set, use the following:

VOLSERNEW=*volsernew*

The VOLUME serial number for a non-cataloged "New" data set.

MEMNEW=*memnew*

The name of a single member in a PDS library, or a member name pattern representing one or more members in the library. You can specify this parameter, or a member name in the DD statement for *ddname*.

A member name pattern can consist of any characters that are valid in a member name, and two special pattern characters: the asterisk (*) and the percent symbol (%).

An * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of *d*, all members in the library whose name contains "d" are processed.

A % is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the library with a 4-character name are processed.

Specification of MEMNEW (or a member in DSNNEW) depends on the parameters used in MEMOLD (or member used in DSNOLD). If MEMOLD (member in DSNOLD) specifies one member, MEMNEW (member in DSNNEW) must also point at one member. If MEMOLD (member in DSNOLD) contains a member name pattern, the specification of MEMNEW (member in DSNNEW) must use the same pattern or an "*".

MEMNEW is ignored if the data set is not a PDS.

MQ:*manager***:***queuename*

You can specify a MQ queue in place of a data set name, where:

managernew
: The “New” MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuenamenew
: The “New” queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

*rt***:***applid***:***rname*

You can specify a CICS resource in place of a data set name, where:

rt
: Resource type. Valid values are:
  FI For a CICS file.
  TD For a Transient Data Queue.
  TS For a Temporary Storage Queue.

applid
: The VTAM applid of the CICS system.

rname
: The name of the resource.

USEIOXNEW

Specifies whether to invoke a user I/O exit, to process the “New” data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the "New" data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXNEW
Specifies the name of the user I/O exit used for the "New" data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXNEW=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXNEW has been set to YES and no installation default has been provided, you must specify IOXNEW=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

BINRECNEW=*binrecnew-len*
Specifies the record length used for processing the "New" HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

KEYNEW=*keynew*
A key for KSDS records or a slot number for RRDS records, for the "New" data set. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record compared. If you omit the *keynew* and *skipnew* values, the comparison begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

SKIPNEW=*skipnew*
Number of logical records to be skipped from the beginning of the "New" data set. The default is 0.

CMPNEW=cmpnew
Number of records from the "New" data set to be compared. The maximum number is 999 999 999. If you specify ALL or omit the parameter, all the remaining records are compared.

New template processing
Use these options to specify the "New" copybook or template that describes the record structure of your "New" data set.

TNEW=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template member that describes the record structure of your "New" data set. The default is TDNEW.

If you specify a concatenated DD, then you must provide the member name, *member*.

TNEWMEM=*member*
The name of the copybook or template member in the datasets identified by the TNEW parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCNEW parameter is specified.

TCNEW=*tcnew***(***tcnmem***)**
PDS and member name of the "New" copybook or template that describes the record structure of your "New" data set.

OFFSETNEW
The length of the 01 field in the “New” template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the “New” template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the “New” template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the “New” template.

FIELDNEW=(*newfield1***,***newfield2***,...)**
Field name or names in the “New” template, used to create a mapping for a formatted comparison. This keyword is used in conjunction with the FIELDOLD keyword to define mapping within the batch file. The parentheses are optional when only one field is specified but mandatory when more than one field is included. The names are mapped in the order given, that is, *oldfield1* is mapped to *newfield1* and so on, overriding any default or existing mapping.

FIELDOLD and FIELDNEW can be used with or without an existing mapping in the template and with or without the CORRESP keyword setting. See Table 14 on page 947 for details.

Copybook processing
If you specify a copybook (instead of an existing template) for either TCOLD or TCNEW, then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL,PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1=RTO1 ... RFROM5=RTO5
Up to five pairs of "From" and "To" pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be

specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

Note: You cannot specify different options for compiling "Old" and "New" copybooks; the same copybook options are used for both.

Comparison options

TYPE=RECORD
Record comparison.

CORRESP=NO
Use this option if the field mapping you want to use is in the TCNEW member. If the TCNEW member is a copybook, or no field mapping is supplied, then File Manager ignores this option and performs a compare as if you had specified CORRESP=YES.

CORRESP=YES
This option instructs File Manager to map output fields to input fields with the corresponding name.

If you want to use the existing mapping in the "New" template, specify CORRESP=NO.

SYNCH=ONETOONE
One-to-one synchronization.

SYNCH=READAHEAD
Read-ahead synchronization. If specified, you can use the following option:

LIMIT=ralim
Limit for read-ahead synchronization.

LENGTH=ralen
The number of records that must match during read-ahead processing for synchronization to occur.

LIST=SUMMARY
Summary listing.

LIST=DELTA
Delta listing.

LIST=MATCHING
Matching listing.

LIST=LONG
Long listing.

LIST=NONE
No listing.

Listing Options
The following option takes effect if the LIST=LONG parameter is specified:

EXCLUDE=*exclude_type*
The specified compare result types are not reported.

exclude_type can have the following values:

INSERTED
Excludes inserted records from the report.

DELETED
Excludes deleted records from the report.

CHANGED
Excludes changed records from the report.

MATCHED
Excludes matched records from the report.

The following options take effect if the LIST keyword is not specified (that is, it defaults to SUMMARY) or is set to anything other than NONE:

NUMDIFF=*numdiff*
The number of differences after which the Compare Utility stops processing the data sets. A *numdiff* value of zero results in all differences being reported.

RCDIFF=(*numdiff***,***return_code***)**
Sets the batch return code when a threshold of changes has been met, where:
numdiff
The minimum number of differences to trigger the return code. A *numdiff* value of zero always trigger the return code.
return_code
The batch return code value that is set if the number of differences have been detected.

WIDE=NO
Narrow listing.

WIDE=YES
Wide listing. The WIDE listing is limited in width to approximately 32K bytes of (record) data. When working with records longer than 32K, the record data, and optional change hilighting, is truncated past 32K bytes of data because of the SYSPRINT output record limitation. The entire record length is used to perform the comparison so records ar marked correctly as "changed" regardless of length.

HEX=NO
No hex formatting.

HEX=YES
Show hex formatting.

Note: The print processing option, DUMP, is ignored and the hexadecimal print output is in updown format.

HILIGHT=NO
No highlighting of changed fields.

HILIGHT=YES
Highlight changed fields.

SHOWATTR=NO
Suppress attribute information in headings (affects formatted comparisons only).

SHOWATTR=YES
Show attribute information in headings (affects formatted comparisons only).

CHNGDFLD=NO
Show all fields in the formatted comparison reports.

CHNGDFLD=YES
Show only changed fields in formatted comparison reports. This option has no effect if the 'Wide listing' (**WIDE=YES**) option has been selected, or for record type comparisons.

SLCTDFLD=NO
Fields selected in the template are not shown in addition to changed fields. This option has effect only if CHNGDFLD=YES is also specified.

SLCTDFLD=YES
Fields selected in the template are always shown. This option has effect only if CHNGDFLD=YES is also specified.

PACK=UNPACK
Allow detection of ISPF packed data and unpack the records if they are packed before passing to the processing routine or printing.

JUST=NO
Do not left-justify numeric fields if Formatted Compare Type option has been selected.

JUST=YES
Left-justify numeric fields if Formatted Compare Type option has been selected

INCLUDEMD
This parameter determines what happens to the descriptor for an Websphere MQ message.

YES The descriptor is added as a prefix as the data is compared.

NO Just the message data is compared. This is the default.

This parameter applies only when the input describes an MQ queue.

PACK=SKIP
Check for ISPF packed data and if packed, skip processing of this data set or member.

Comparison output data sets

IOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the "inserted" records are to be written to.

IOUTMEM=*member*
The name of the member in the dataset identified by the COUTPUT parameter if it has not been specified on the DD statement.

IOUTDSN=*ioutdsn(member)*
The name of the output data set, or an absolute path to the output HFS file, the "inserted" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

IBINREC=*binrec_len*
Specifies the record length to be used for processing the inserted HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

IDISP=MOD
Writes inserted records to the existing output data set, starting from the beginning.

IDISP=OLD
Appends inserted records to the existing output data set.

DOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the "deleted" records are to be written to.

DOUTMEM=*member*
The name of the member in the dataset identified by the COUTPUT parameter if it has not been specified on the DD statement.

DOUTDSN=*doutdsn(member)*
The name of the output data set, or an absolute path to the output HFS file, the "deleted" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

DBINREC=*binrec_len*
Specifies the record length to be used for processing the deleted HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

DDISP=MOD
Writes deleted records to the existing output data set, starting from the beginning.

DDISP=OLD
Appends deleted records to the existing output data set.

NCOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the new "changed" records are to be written to. The default is FMNCHOUT.

NCOUTMEM=*member*
The name of the member in the dataset identified by the NCOUTPUT parameter if it has not been specified on the DD statement.

NCOUTDSN=*ncoutdsn(member)*
The name of the output data set, or an absolute path to the output HFS file, the new "changed" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

NCBINREC=*binrec_len*
Specifies the record length to be used for processing the New Change HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

NCDISP=MOD
Writes new changed records to the existing output data set, starting from the beginning.

NCDISP=OLD
Appends new changed records to the existing output data set.

OCOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the new "changed" records are to be written to. The default is FMOCHOUT.

OCOUTMEM=*member*
The name of the member in the dataset identified by the OCOUTPUT parameter if it has not been specified on the DD statement.

OCOUTDSN=*ocoutdsn(member)*
The name of the output data set, or an absolute path to the output HFS file, the old "changed" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

OCBINREC=*binrec_len*
Specifies the record length to be used for processing the New Change HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

OCDISP=MOD
Writes old changed records to the existing output data set, starting from the beginning.

OCDISP=OLD
Appends old changed records to the existing output data set.

NMOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the new "matched" records are to be written to. The default is FMNMTOUT.

NMOUTMEM=*member*
The name of the member in the dataset identified by the COUTPUT parameter if it has not been specified on the DD statement.

NMOUTDSN=*nmoutdsn(member)*
The name of the output data set, or an absolute path to the matched HFS file, the new "matched" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

NMBINREC=*binrec_len*
Specifies the record length to be used for processing the new matched HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

NMDISP=MOD
Writes new matched records to the existing output data set, starting from the beginning.

NMDISP=OLD
Appends new matched records to the existing output data set.

OMOUTPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data set or HFS file the old "matched" records are to be written to. The default is FMOMTOUT.

OMOUTMEM=*member*
The name of the member in the dataset identified by the COUTPUT parameter if it has not been specified on the DD statement.

OMOUTDSN=*omoutdsn(member)*
The name of the output data set, or an absolute path to the matched HFS file, the old "matched" records are to be written to. The name can include a member name in parenthesis.

An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line.

OMBINREC=*binrec_len*
Specifies the record length to be used for processing the old matched HFS output data set. Valid range is 1–32760. The file is processed in binary mode. If you do not specify this parameter, the file is processed in text mode.

OMDISP=MOD
Writes old matched records to the existing output data set, starting from the beginning.

OMDISP=OLD
Appends old matched records to the existing output data set.

Formatted type options
If you specify TYPE=FORMATTED, for a formatted comparison, you can use the following options:

IGNORELB=NO
Respect leading blanks when comparing alphanumeric fields.

IGNORELB=YES
Ignore leading blanks when comparing alphanumeric fields.

IGNORETB=NO
Respect trailing blanks when comparing alphanumeric fields.

IGNORETB=YES
Ignore trailing blanks when comparing alphanumeric fields.

MATCHREF=NO
Ignore leading blanks (unless IGNORELB=NO is also specified), trailing blanks (unless IGNORETB=YES also specified) and embedded blanks when comparing alphanumeric fields.

MATCHREF=YES
Respect leading blanks, trailing blanks and embedded blanks when comparing alphanumeric fields.

Note: Any setting of IGNORELB or IGNORETB is ignored if MATCHREF=YES is also specified.

IGNORECASE=NO
Respect case when comparing alphanumeric fields.

IGNORECASE=YES
Ignore case when comparing alphanumeric fields.

Keyed synchronization options
If you specify SYNCH=KEYED, you are requesting keyed synchronization.

If specified, you can define up to sixteen key segments to be concatenated to form a single key. A key segment is comprised of the key's location in the “Old” and “New” data sets and the key's length and data type. These values are built using one of the following sources or a combination of these elements:

- The intrinsic data set keys (where the data set is keyed, for example, VSAM KSDS).

  If only one of the data sets is keyed, the location and length values for the first segment is initialized from the available data set key information, and the keywords corresponding to the initialized values are not required. However, you need to supply the location value and, optionally, the data type for the non-keyed data set. If you do not specify the data type, the default type of AN is used.

  If both data sets are keyed, you do not need to supply any of the keywords and the location values is initialized from the data set key. The length value is that of the shorter of the two keys.

If used in a multi-segment key, this defines the first segment. To prevent other segment sources from overriding the intrinsic key, leave the first argument in the **KEYxxx** keyfields as a void, for example, KEYLOCOLD=(,*keylocold*).

- The key segment sequence information stored in your templates.
- The **KEYFLDOLD** and **KEYFLDNEW** keyfields (when a template has been specified)
- The **KEYLOCOLD**, **KEYLOCNEW**, **KEYLEN**, and (optionally) **KEYTYPE** keyfields.

In most situations, you would use only one of these sources to define your key segments, however, all of them can be intermixed to define the key segments, with the caveat that **KEYLOCOLD**, **KEYLOCNEW**, **KEYLEN**, and **KEYTYPE** override **KEYFLDOLD** and **KEYFLDNEW** when both specify data for the same segment.

Up to 16 values can be specified for each keyword, and the parentheses can be omitted if only one value is provided. The same number of key segments must be defined for both data sets, and the data type and length of corresponding segments must match. Each set of corresponding values then defines a key segment.

Note:

1. **KEYFLDOLD** and **KEYFLDNEW** can only be specified if a template is specified for the corresponding data set.
2. Any segment whose presence is implied by a keyword must be fully specified, except that the data type for the segment can default as described in **KEYTYPE** below.

KEYTCOLD

Determines whether or not any key segment information stored in the “Old” template is used.

If unspecified, the default behavior is that key segment information stored in the “Old” template is only used when all of the following conditions are met:

- TCOLD=*tcold* has been specified.
- The “Old” template contains key segment information.
- No other keyed synchronization options have been specified for either the “Old” or the “New” data sets.

If any of these conditions are not met, key segment information in an “Old” template is ignored.

If specified, the setting given overrides the default behavior.

YES The key segment information in the “Old” template is loaded regardless of whether or not other **KEY...** keywords have been specified. Where present, the other **KEY...** keywords function as overrides to the template specifications.

NO The key segment information in the “Old” template is ignored regardless of whether or not other **KEY...** keywords have been specified.

KEYTCNEW
Determines whether or not any key segment information stored in the "New" template is used.

If unspecified, the default behavior is that key information stored in the "New" template is only used when all of the following conditions are met:

- TCNEW=*tcnew* has been specified.
- The "New" template contains key segment information.
- No other keyed synchronization options have been specified for either the "Old" or the "New" data sets.

If any of these conditions are not met, key segment information in an "New" template is ignored.

If specified, the setting given overrides the default behavior.

YES The key segment information in the "New" template is loaded regardless of whether or not other **KEY...** keywords have been specified. Where present, the other **KEY...** keywords function as overrides to the template specifications.

NO The key segment information in the "New" template is ignored regardless of whether or not other **KEY...** keywords have been specified.

KEYFLDOLD=(*keyfieldold***,...)**
Specifies the name of the field or fields in the "Old" template to be used as key segments. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keyfieldold* value or values in the concatenated key.

KEYFLDNEW=(*keyfieldnew***,...)**
Specifies the name of the field or fields in the "New" template to be used as key segments. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keyfieldnew* value or values in the concatenated key.

KEYLOCOLD=(*keylocold***,...)**
Key locations in "Old" data set for keyed synchronization. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keylocold* value or values in the concatenated key.

KEYLOCNEW=(*keylocnew***)**
Key location in "New" data set for keyed synchronization. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keylocnew* value or values in the concatenated key.

KEYLEN=keylen
Key length for keyed synchronization. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keylen* value in the concatenated key.

KEYTYPE
Defines the data type of the key segment. Can be used to override

the data type of an existing template field. When used in combination with other key segment sources, empty arguments must be included to indicate the position of the *keylen* value in the concatenated key.

KEYTYPE is optional - any key segments defined by **KEYLOCxxx/KEYLEN** that do not have a corresponding **KEYTYPE** keyword is given type **CHAR**.

Key segments defined using **KEYFLDOLD** and **KEYFLDNEW** inherit the type of the template field on which they are based, and the expected record sequence is assumed to follow accordingly. In practice this is only likely to be significant for signed binary and packed decimal fields (though internal floating point is also supported). For example, if a key were defined on a two-byte field by location and length, then a value of '001C'x would be less than a value of '001D'x. However if the key were defined via a packed-decimal template field then '001D'x (-1) would be less than '001C'x (+1). Equivalent results can be obtained by using **KEYTYPE** with **KEYLOCxxx** and **KEYLEN**.

CHAR
This is equivalent to the internal C/AN data type. CHAR data type segments are synchronized using the normal EBCDIC collating sequence.

BINARY
This is equivalent to the internal B/BI data type. BINARY data type segments are synchronized as signed binary integers. They must have a length of 2, 4 or 8.

PACKED
This is equivalent to the internal P/PD data type. PACKED data type segments are synchronized as signed packed decimal integers. They must have a length less than or equal to 16.

FLOAT
This is equivalent to the internal FP data type. FLOAT data type segments are synchronized as signed floating point numbers. They must have a length less of 4 or 8.

Note: Template fields that are selected as key segments but do not have one of the above data types, is treated as CHAR. In particular, this means that the actual data lengths of varying fields is ignored.

Load module compare options
If you specify SYNCH=LMOD, you are requesting load module comparison.

Load module comparison has these specific options:

CMPLVL
Determines the level of load module comparison.

LMOD
Only information on the load module level is extracted and compared. CSECT information (and differences at CSECT level) is ignored. This results in a less detailed comparison.

CSECT
Information on both the load module and CSECT levels is extracted and compared. This results in a detailed comparison.

DATEFORM=YYYYDDD
Reported dates (link and compile dates) shown in YYYY.DDD format.

DATEFORM=YYMMDD
Reported dates (link and compile dates) shown in YY/MM/DD format.

LMODC
Determines what information at the load module level is to be included in the compare. The criteria correspond with load module properties; only those specified are compared and displayed. Each of the options below can be specified in any sequence, enclosed in parenthesis:

SIZE The load module size is compared.

ADDRESS
The entry point address of load module is compared.

LINKER | BINDER
The version of the linkage editor or binder used to prepare the load module is compared. LINKER and BINDER are mutually exclusive.

DATE | TIME
The load module link (bind) date and time are compared. DATE and TIME are mutually exclusive.

MODE
The AMODE and RMODE of the load module are compared.

AC | AUTH
The load module authorization code is compared. AC and AUTH are mutually exclusive.

ATTR The load module link (bind) attributes are compared.

CSECTC
Determines what information at the CSECT level is to be included in the compare. The criteria correspond with the CSECT properties; only those specified are compared and displayed. Each of the options below can be specified in any sequence, enclosed in parenthesis. If CMPLVL=LMOD, this parameter is ignored.

SIZE The CSECT size is compared.

ADDRESS
The address of the CSECT is compared.

COMPILER
The versions of the language compilers used to compile the CSECT are compared

DATE The date of the CSECT compile is compared.

MODE
The AMODE and RMODE of the CSECT are compared.

IDRZAP | ZAP
The AMSPZAP IDR data is compared. The IDR ZAP data is an extension of the CSECT information, but is formatted into separate records. ISRZAP and ZAP are mutually exclusive.

TEXT The CSECT content is compared. The CSECT content is an extension of the CSECT information, but is formatted into separate, 32-byte records shown in "memory dump" format (hexadecimal and character).

Template reporting options

PBK=NONE
Template layout and criteria information not reported.

PBK=LAYOUTS
Template layout information reported if TYPE=FORMATTED is also specified.

PBK=CRITERIA
Template criteria information reported. In this case, a terse layout report is also generated, containing only the fields referred to in the criteria expressions.

PBK=ALL
Template layout and criteria information reported. If TYPE=RECORD is specified, a terse layout report is generated, containing only the fields referred to in the criteria expressions. If TYPE=FORMATTED is specified, a full layout report is generated.

MAP=NONE
Template mapping information not reported.

MAP=MAPPED
Mapped template fields reported if TYPE=FORMATTED is also specified.

MAP=UNMAPPED
Unmapped template fields reported if TYPE=FORMATTED is also specified.

MAP=ALL
Both mapped and unmapped template fields reported if TYPE=FORMATTED is also specified.

ARRAY=YES
If field information is reported, all occurrences of any array elements are reported.

For COBOL OCCURS DEPENDING ON tables, the maximum occurrences are reported.

For PL/1 REFER arrays, the minimum occurrences are reported.

ARRAY=NO
Occurrences of array elements not reported.

FSTATS=NO
The Field Statistics report is not to be generated.

FSTATS=YES
The Field Statistics report is to be generated.

HEXLOC=YES
If field information is reported, all field start and end positions are reported as hexadecimal offsets.

HEXLOC=NO
Field start and end positions not reported as hexadecimal offsets.

HEXLEN=YES
If field information is reported, all field lengths are reported in hexadecimal.

HEXLEN=NO
Field lengths not reported in hexadecimal.

```
//DSM JOB (ACCT),'NAME'
//* COMPARE DATA SETS
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//STPSSEX     EXEC FMBAT
//SYSIN    DD *
$$FILEM VER
$$FILEM DSM TYPE=RECORD,
$$FILEM     SYNCH=ONETOONE,
$$FILEM     LIST=LONG,
$$FILEM     WIDE=YES,
$$FILEM     HILIGHT=YES,
$$FILEM     DSNOLD=FMN.SFMNSAM1(FMNCDATA),
$$FILEM     SKIPOLD=0,
$$FILEM     CMPOLD=20,
$$FILEM     DSNNEW=FMN.SFMNSAM1(FMNCDATA),
$$FILEM     SKIPNEW=20,
$$FILEM     CMPNEW=20
$$FILEM     EOJ
/*
$$FILEM DSCMP TYPE=RECORD,
$$FILEM SYNCH=KEYED,
$$FILEM LIST=LONG,
$$FILEM WIDE=YES,
$$FILEM HILIGHT=YES,
$$FILEM HEX=YES,
$$FILEM DSNOLD=USERID.COMPARE.KSDS1,[1]
$$FILEM KEYLOCOLD=(,11),[2]
$$FILEM DSNNEW=USERID.COMPARE.FLAT2,[3]
$$FILEM TCNEW=USERID.TEMPLATE(FLAT2),[4]
$$FILEM KEYTCNEW=YES,[5]
$$FILEM KEYFLDNEW=(,FIELD-3),[6]
$$FILEM KEYLEN=(,1),[7]
$$FILEM KEYTYPE=(,CHAR)[8]
$$FILEM     EOJ
```

Note:

1. DSNOLD is intrinsically keyed with key position 1 and length 10.
2. KEYLOCOLD provides a position for the second segment of the “Old” key.
3. DSNNEW is not intrinsically keyed.
4. TCNEW specifies a new template: the template specifies FIELD-1 (position 1, length 10, type AN) as key segment 1.

5. KEYTCNEW forces the loading of the TCNEW key segment information, in spite of the presence of the other KEY... keywords, which act as overrides wherever they clash with the information from the template.
6. KEYFLDNEW specifies FIELD-3 (position 12, length 2, type BI) as the second segment of the "New" key
7. KEYLEN overrides the length of the second key segment
8. KEYTYPE overrides the type of the second key segment

The resulting key segments are shown in the following table:

Segment number	"Old" key position	"New" key position	Key length	Key type
1	1[1]	1[2]	10[3]	AN[4]
2	11[5]	12[6]	1[7]	AN[8]

Source of key information:

1. DSNOLD catalog entry
2. Template field FIELD-1
3. DSNOLD catalog entry and template field FIELD-1 (must be consistent)
4. Template field FIELD-1 (consistent with AN default for key type)
5. KEYLOCOLD
6. Template field FIELD-3 via KEYFLDNEW
7. KEYLEN
8. KEYTYPE

DSP (Data Set Print)

Purpose

Use the DSP function to print sequential data sets, VSAM data sets, PDS members, HFS files, MQ queues, or CICS resources in a selected format. You can print data by:

- record
- block (non-VSAM)
- control interval (VSAM)
- field (if a template or copybook has been provided)

You can select records for printing using:

- Member name selection criteria
- Date created selection criteria
- Date last modified selection criteria
- User ID selection criteria
- the start key (VSAM only)
- skip and print count fields
- conditional expression defined in the provided template

The print function can be run in the foreground or as a batch job. The output for the print function is controlled by the SET options.

Usage notes

Choose between four data print formats:

- character
- hexadecimal
- single-record, using a template
- multiple-record (tabular), using a template

When you use a copybook or template, records are formatted field by field using the record layout defined in the copybook or template.

You can Print concatenated data sets with like or unlike attributes. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for processing. Such invocation may fail as DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

Performance tips

- See "General tips about performance when you use File Manager functions" on page 833. The comments about File Manager using DFSORT technology when performing sequential file I/O are important to DSP performance.

Options

You can specify the following options:

- Whether to process logical records or physical blocks.
- The position of the first record to print.
- The number of records to print.
- The name of a DFSORT or REXX procedure, if you want the output records passed to a procedure for processing before they are printed.

Physical block processing is not compatible with SNGL or TABL print format, or with using templates, and SNGL or TABL print format requires you to use a template.

You can use the various SET processing options to control the print output:

- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- When you specify CHAR or LHEX print format, SET RECLIMIT controls how many bytes of each record are printed, and SET DATAHDR determines whether header information, such as record number, is printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Note that additional formatting options are available when using formatted print in SNGL mode. Additional field information (redefined fields, field reference number, field type and length values, picture clause, start location, structure and numeric field justification) can be set and printed depending on the options selected using the keywords listed later in this section.

When you specify the PROC option, you are supplying a REXX procedure. For more information, see the *proc* parameter below.

Return codes

The default return codes from the DSP function have the following modified meanings:

1 No records printed for some of multiple members

2 Print error encountered

3 REXX member selection is in effect but the procedure encountered

a RETURN DROP, STOP or STOP IMMEDIATE string. This has been treated as a RETURN string with no arguments. OR REXX member selection is NOT in effect but the procedure encountered a RETURN DROP MEMBER or RETURN PROCESS MEMBER string. This has been treated as a RETURN string with no arguments.

4 No records printed because no records selected

4 No records printed because no members to process

4 No records printed because input empty

4 No records printed because the input data set or member is in ISPF Packed Data format and the "PACK=SKIP" option was specified

8 REXX non-syntax error encountered while processing records

16 Data set or member open error

16 Data set not found

16 Input data appears ISPF packed but is not valid.

16 Other input or output error occurred

16 Insufficient storage available

16 DSP abended

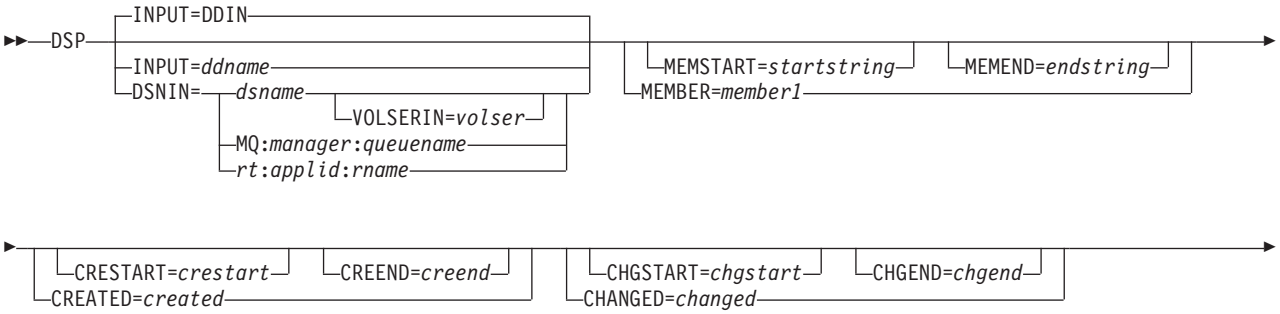
16 Other serious error that stops processing occurred

Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

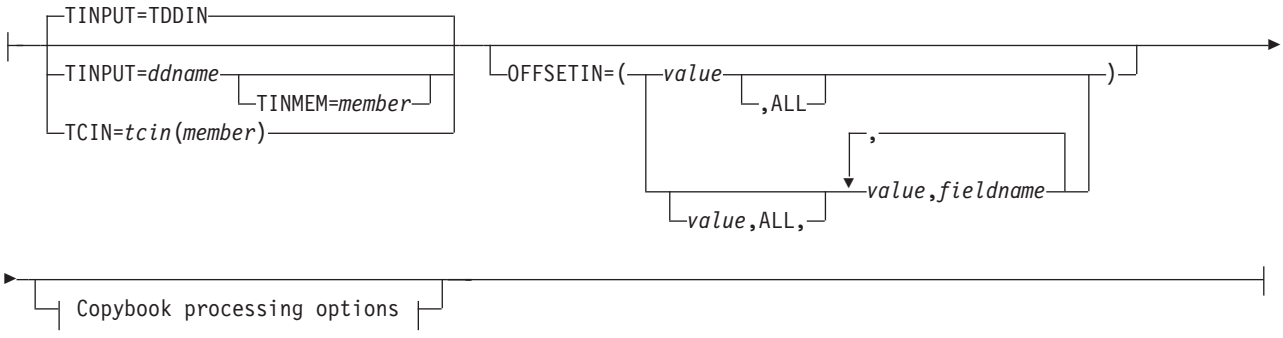
Related functions

DP Print physical disk records
DVT Print VTOC entries

Syntax

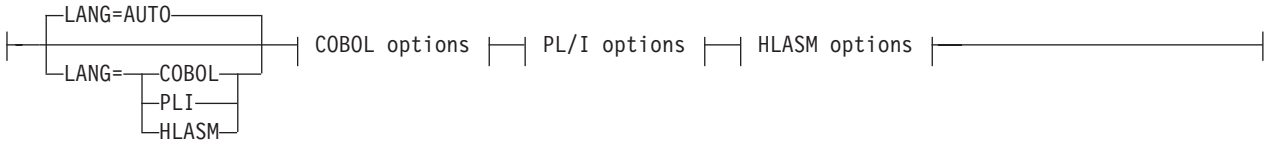


Template processing options:

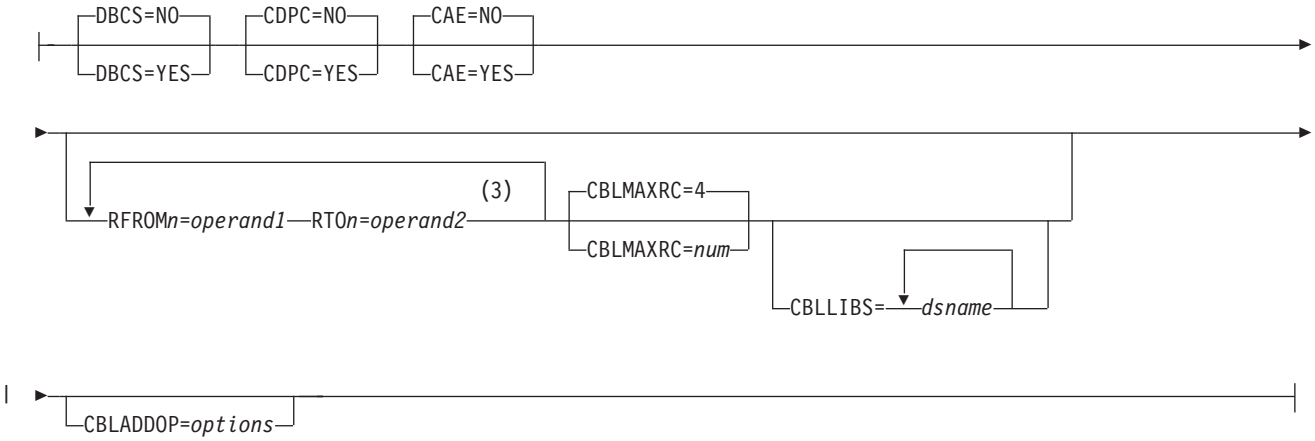


Copybook processing options:

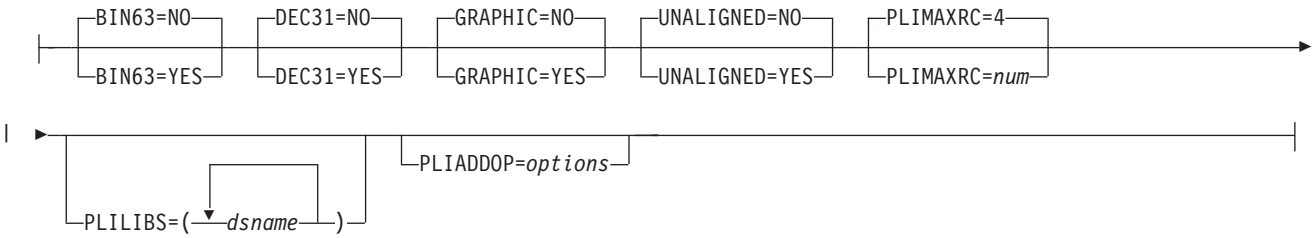
Function reference: DSP



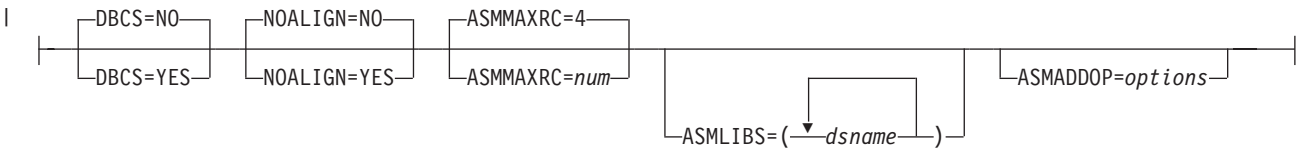
COBOL options:



PL/I options:



HLASM options:



Notes:

1 VSAM only.

2 Non-VSAM only.

3 *n*= 1 to 5.

INPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the input data set or HFS file. The default is DDIN.

DSNIN=*dsname*
Defines the name of the input data set or an absolute path to the input HFS file (directory). If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. An absolute path to an HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, you can split it over more than one line. You can further describe this data set, as follows:

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

MEMBER=*member1*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the input data set is a PDS(E), you may specify this parameter, or a member name in the DD statement for *ddname*, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

***** represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%`, all members in the PDS whose name is four characters in length are processed.

member1 is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the copy. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the copy. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

manager
The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuename
The queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

rt:*applid*:*rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the copy.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the copy.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00').

If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

USEIOXIN
Specifies whether to invoke a user I/O exit, to process the input data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the input data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXIN
Specifies the name of the user I/O exit used for the input data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXIN=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXIN has been set to YES and no installation default has been provided, you must specify IOXIN=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

BINRECIN=*binrecin-len*
Specifies the record length used for processing the HFS file. Valid range: 1 to 32760.

The file is processed in Binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in Text mode (variable-length records, boundaries determined by delimiters).

CIACCESS=NO
Process logical records.

CIACCESS=YES
Process control intervals.

BLKACCESS=NO
Process logical records.

BLKACCESS=YES
Process blocks.

CCSID=YES
Show the CCSID assigned to each field, if applicable, on SNGL print.

CCSID=NO
Do not show the CCSID assigned to each field, if applicable, on SNGL print.

FORMAT=*format*
The format of the output:
CHAR
Character format (the default).
HEX Hexadecimal format.
SNGL Single-record format (one field on each line). This option is available when viewing logical records.
TABL Tabular format (fields printed across the page). This option is available when viewing logical records.

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

SMPLINCL=*sample_include*
Number of physical records to be included in a repeating sample from a data set. Both SMPLINCL and SMPLSKP keywords must have non-zero values for sampling to take effect. Range: 0–9999999.

SMPLSKP=*sample_skip*
Number of physical records to be skipped in a repeating sample from a data set. Both SMPLINCL and SMPLSKP keywords must have non-zero values for sampling to take effect. Range: 0–9999999.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record printed. If you omit the *key* and *skip* values, printing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

NLRECS
Number of records to be printed or ALL.

ALL If you specify ALL or omit the parameter, all the remaining records are copied.

nlrecs The maximum number is 999 999 999.

PACK Determines if File Manager should detect if the input data is in ISPF packed format. This keyword is ignored when processing VSAM data sets. When an I/O exit has been specified for either the input or output data set (or both), the only valid option is PACK=NONE.

UNPACK
Instructs File Manager to check if the input data set is in ISPF packed format and if so, to unpack it before print processing.

NONE
Instructs File Manager not to check if the input data set is in ISPF packed format.

SKIP Instructs File Manager to determine if the input data set is in ISPF packed format and if so, to skip the print processing.

PROC=*proc*
Member name of a REXX procedure that you want to use to process each record before it is printed, or an asterisk (*) to indicate the REXX procedure is inline. If you specify a member name, you must define an FMNEXEC

ddname that identifies the PDS containing the member. If you specify *, the procedure is read from SYSIN immediately following the control statement for the current function. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

For more information about using REXX procedures to process records before they are printed, see Chapter 13, “Enhancing File Manager processing,” on page 405.

MEMPROC
Specifies that REXX member selection is in effect. Records are read from the input member and then cached in memory until a decision is made, within the REXX procedure, on whether the member is to be copied or dropped. Once the decision has been made, the entire member is either copied or dropped, depending upon the RETURN string specified in the REXX procedure. If the entire member is processed without encountering a RETURN DROP MEMBER or RETURN PROCESS MEMBER string, the member is processed according to the action specified by the parameter specified for MEMPROC. These are:

PROCESS
The member is to be included in the copy. The member is copied intact, subject to any specified template processing, which is performed before the user REXX proc is invoked. This is the default action, if no parameter is specified with the MEMPROC keyword.

DROP The member is to be excluded from the copy. Processing continues with the next member.

REF=YES
Show field reference number on SNGL print.

REF=NO
Do not show field reference number on SNGL print.

RDF=YES
Show redefined fields on SNGL or TABL print.

RDF=NO
Do not show redefined fields on SNGL or TABL print.

PIC=YES
Show picture clause on SNGL print.

PIC=NO
Do not show picture clause on SNGL print.

SLOC=YES
Show start location on SNGL print.

SLOC=NO
Do not show start location on SNGL print.

STR=YES
Show structure on SNGL print.

STR=NO
Do not show structure on SNGL print.

JUST=YES
Left-justify numeric fields on SNGL print.

JUST=NO
Do not left-justify numeric fields on SNGL print.

TYPE=YES
Show field type and length values on SNGL print.

TYPE=NO
Do not show field type and length values on SNGL print.

RLEN=YES
Print record length in TABL and SNGL formats.

RLEN=NO
Do not print record length in TABL and SNGL formats.

IGNLEN
Specifies whether or not File Manager ignores length mismatches when selecting records for processing.

NO Do not ignore length mismatches. Records that are shorter than the matching structure length in the template are not selected for processing.

YES Use this option to ignore length mismatches.

INRDW
Controls whether or not to adjust the input start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the input start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the input record.

OUTRDW
Controls whether or not to adjust the output start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the output start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the output record.

MEMLIST
Allows you to specify a list of member names.

member_n
The name of the member to be processed. Generic name masks are allowed.

INCLUDEMD
This parameter determines what happens to the descriptor for an Websphere MQ message.

YES The descriptor is added as a prefix as the data is printed.

NO Just the message data is printed. This is the default.

This parameter applies only when the input describes an MQ queue.

Template processing
Define which template (if any) is used to describe the record structure in the input data set, and how File Manager processes this template.

TINPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your input data. The default is TDDIN.

If you specify a concatenated DD, then you must provide the member name, *member*.

TINMEM=*member*
The name of the copybook or template member in the datasets identified by the TINPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCIN parameter is specified.

TCIN=*tcin***(***member***)**
PDS and member name of the copybook or template that describes the record structure of your input data.

OFFSETIN
The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

PACK Determines if File Manager should detect if the input data sets are in ISPF packed format. This keyword is ignored when processing

VSAM data sets. When an I/O exit has been specified for either data set (or both), the only valid option is PACK=NONE.

UNPACK
Instructs File Manager to check if the input data sets are in ISPF packed format and if they are, to unpack them before the comparison.

NONE
Instructs File Manager not to check if the input data sets are in ISPF packed format.

SKIP Instructs File Manager to check if the input data set is in ISPF packed format and if so, to skip the compare processing.

```
//DSP JOB (acct),'name'  Print QSAM Data
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//STPSPEX  EXEC FMBAT
//SYSIN    DD *
$$FILEM VER
$$FILEM DSP DSNIN=SYS1.PROCLIB,MEMBER=COBUCLG
$$FILEM EOJ
/*
```

```
//DSPJPN JOB  (acct),'name'  Print with DBCS characters
//JAPEF96 OUTPUT DUPLEX=NORMAL,CHARS=(GT15,EF96),PRMODE=SOSI1
//FILEMGR  EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*,OUTPUT=(*.JAPEF96)
//FMNTSPRT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM SET LANGUAGE=JAPANESE
$$FILEM DSP FORMAT=SNGL,
$$FILEM     TCIN=hlq.TEMPLATE(member),
$$FILEM     DSNIN=hlq.DBCSDATA
```

```
//FMNUSR3 JOB  (FMNUSER),'USER',USER=FMNUSER,NOTIFY=FMNUSER,
//  TIME=(5),CLASS=A,MSGLEVEL=(1,1),MSGCLASS=H
//*
//* TEST PRINTING USING JAPANESE CHARACTER SETS WITH FMT
//*
//JAPEF96 OUTPUT DUPLEX=NORMAL,CHARS=(GT15,EF96),PRMODE=SOSI1
//FILEMGR  EXEC PGM=FMNMAIN
//STEPLIB  DD DSN=FMN.V2R1M0.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=H,OUTPUT=(*.JAPEF96),DEST=(PTHMVS8,QAPT22Q1)
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM SET LANGUAGE=JAPANESE
$$FILEM FMT SET,FLD=(11,20,DB),FLD=(31,40,DB),FLD=(51,60,DB),
$$FILEM     FLD=(71,80,DB)
$$FILEM DSP DSNIN=FMNUSER.JPN.TESTDATA,MEMBER=$FMTDATA
/*
```

DSU (Data Set Update) — batch only

Purpose
Update disk data set records.

Usage notes

Use this function to update logical records in a single sequential disk data set, a single VSAM data set, one or more members of a PDS, an MQ queue, or a CICS resource.

Note: DFSMS-compressed datasets are not supported (for use with DSU).

You can select the records to be processed using:

- Member name selection criteria
- Date created selection criteria
- Date last modified selection criteria
- User ID selection criteria

Records in the data set are read sequentially. After each record is read, File Manager invokes the REXX procedure specified in the PROC parameter, and passes the contents of the record to the exec. The contents are passed in two File Manager-defined REXX variables, INREC and OUTREC. When the exec is invoked, the contents of the two variables are identical. The INREC variable is intended to be used as a reference variable. Any changes made to it are ignored by File Manager. The OUTREC variable can be updated by the exec. After the REXX procedure has processed the record, if the data in OUTREC has changed, the record is updated in the data set using the contents of OUTREC.

You cannot add records or delete records using DSU. If you need to add or delete records, you can use one of the File Manager data set copy functions. You cannot change the length of records in a data set using DSU. If the REXX procedure increases the length of the data in OUTREC, the data is truncated to its original length before the record is updated. If the REXX procedure decreases the length of the data in OUTREC, the data is padded to its original length using the pad value specified in the PAD processing option. If no pad value has been specified, the contents of the record are unpredictable.

Performance tips

- When you use DSU to update members of a PDS(E):

  One DSU default is STATS=ON, which causes the ISPF statistics for each updated member to be updated. This can significantly increase I/O (EXCP) and CPU utilization. To improve performance, consider using STATS=OFF.

Options

When you specify the PROC option, you are supplying a REXX procedure. For more information, see the *proc* parameter below.

Return codes

The default return codes from the DSU function have the following modified meanings:

1 One or more members not updated

2 Change failed (for example invalid key change)

4 No records updated (NOUPDATE=NO)

4 No records processed because no members to process

4 No records processed because input empty

4 No records processed because input is in ISPF Packed Data format and the "PACK=STOP" option was specified.

8 REXX non-syntax error encountered while processing records

16 Program Object specified - this is not supported

16 Data set or member in use

16 Data set or member open error

16 Data set not found

16 Other input or output error occurred

16 Insufficient storage available

16 DSU abended

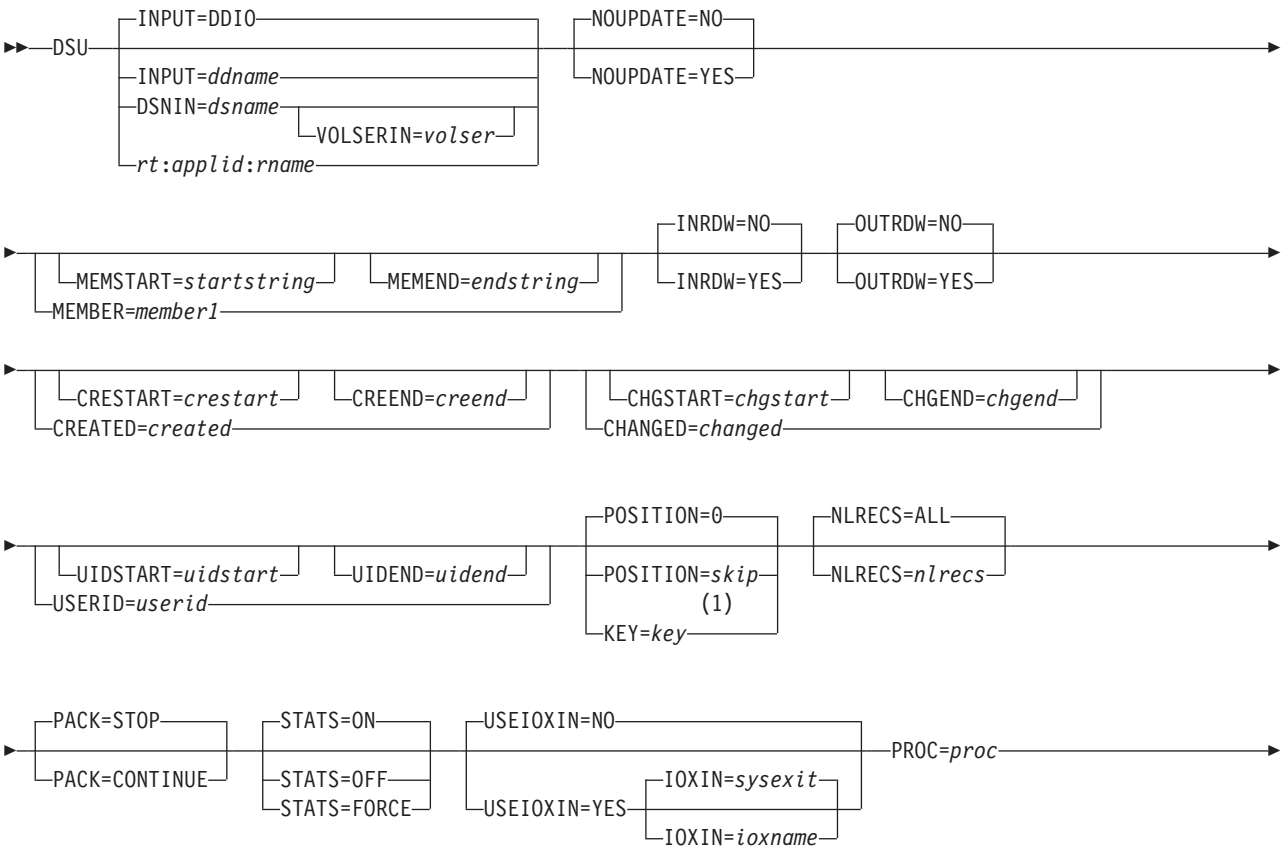
16 Other serious error that stops processing occurred

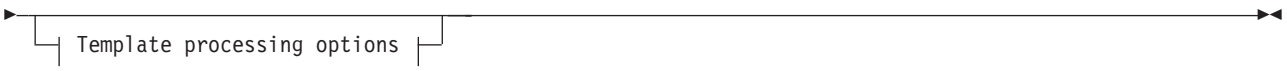
Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Related functions

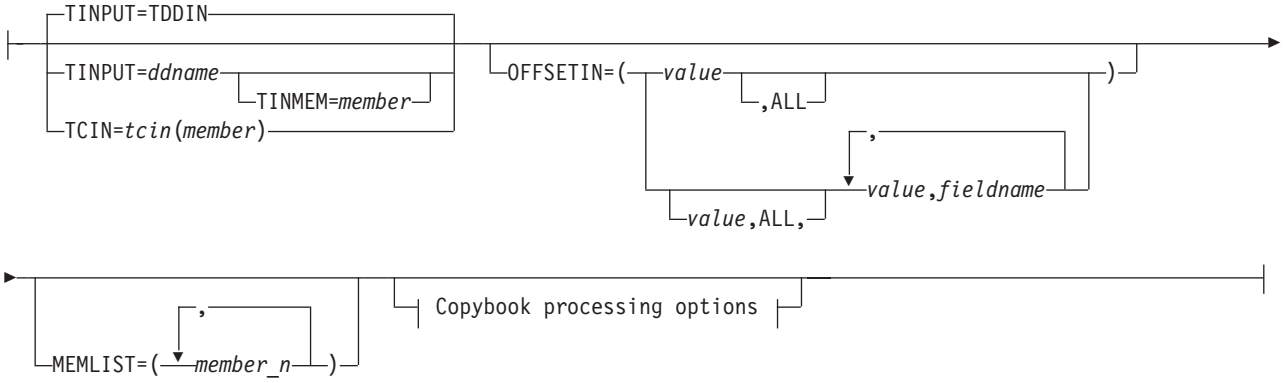
DSEB Edit a data set via batch job processing.
DSX Display the extents of a data set.

Syntax

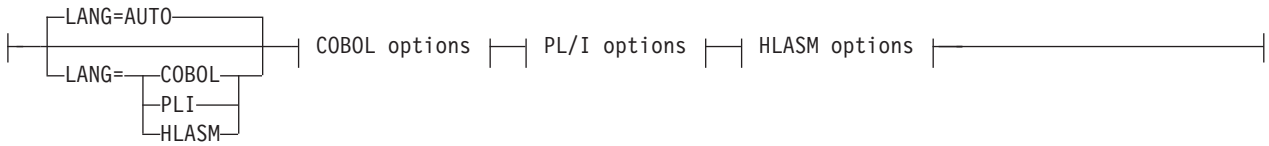




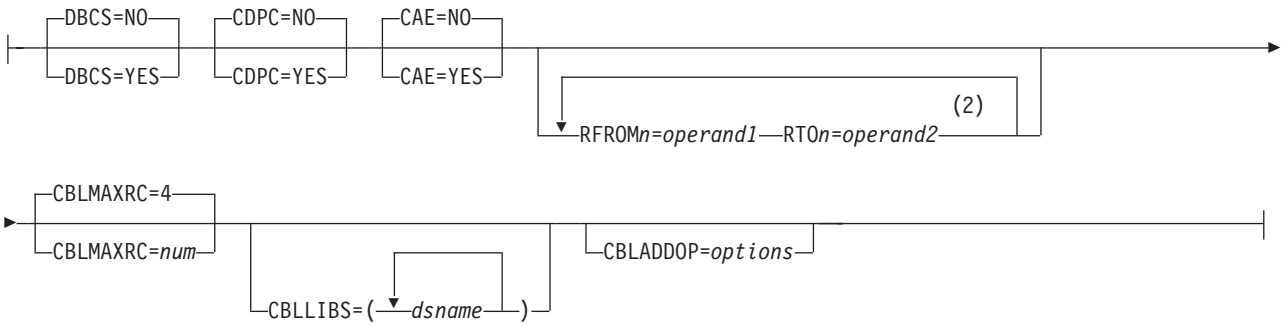
Template processing options:



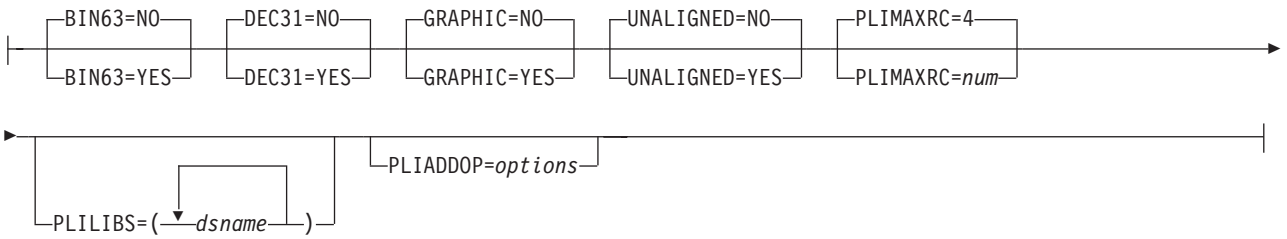
Copybook processing options:



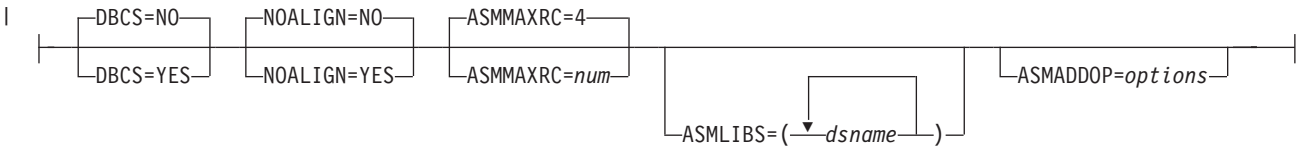
COBOL options:



PL/I options:



HLASM options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

INPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the input data set. The default is `DDIN`.

DSNIN=*dsname*
Defines the name of the input data set. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. You can further describe this data set, as follows:

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

rt:*applid*:*rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

NOUPDATE
Allows you to specify that you intend no updates to the data set while executing the utility.

NO Updates to the data are honored.

YES Forces the allocation of the data set as input only. All updates to the data are ignored.

MEMBER=*member1*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the input data set is a PDS(E), you may specify this parameter, or a member name in the DD statement for *ddname*, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if

you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%`, all members in the PDS whose name is four characters in length are processed.

member1 is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the copy. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the copy. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

INRDW
Controls whether or not to adjust the input start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the input start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the input record.

OUTRDW
Controls whether or not to adjust the output start location when the specified start location takes into account the record descriptor word (RDW).

NO Does not adjust the output start location.

YES Subtracts 4 from all start locations that have been coded on external functions that refer to the output record.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:
DD = 31
MM = 12

YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the copy.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the copy.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00'). If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

KEY=*key* **(VSAM only)**
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record updated. If you omit the *key* and *skip* values, updating begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

NLRECS
Number of records to be printed or ALL.

ALL If you specify ALL or omit the parameter, all the remaining records are copied.

nlrecs The maximum number is 99 999 999.

PACK Determines if File Manager should detect if the input data is in ISPF packed format.

STOP Default. File Manager detects whether the input data is in ISPF packed format, and if it is, stops the processing.

CONTINUE
File Manager does not detect whether the input data is in ISPF packed format and continues processing.

STATS=ON
Default. This updates the ISPF statistics (if already present) when a PDS or PDSE member has been changed.

STATS=OFF
The ISPF statistics is not updated when a PDS or PDSE member has been changed.

STATS=FORCE
The ISPF statistics that exist for members being processed are always updated and statistics for a member that previously did not have statistics are created.

USEIOXIN
Specifies whether to invoke a user I/O exit, to process the input data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the input data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXIN
Specifies the name of the user I/O exit used for the input data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXIN=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXIN has been set to YES and no installation default has been provided, you must specify IOXIN=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

PROC=*proc*
Member name of a REXX procedure that you want to use to process each record before it is updated, or an asterisk (*) to indicate the REXX procedure is inline. If you specify a member name, you must define an FMNEXEC ddname that identifies the PDS containing the member. If you specify *, the procedure is read from SYSIN immediately following the control statement for the current function. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

For more information about using REXX procedures to process records before they are updated, see Chapter 13, “Enhancing File Manager processing,” on page 405.

Template processing

Define which template (if any) is used to describe the record structure in the input data set, and how File Manager processes this template.

TINPUT=*ddname*

Defines a reference to a DD or TSO ALLOC statement for the data sets which contain the copybook or template that describes the record structure of your input data. The default is TDDIN.

If you specify a concatenated DD, then you must provide the member name, *member*.

TINMEM=*member*

The name of the copybook or template member in the datasets identified by the TINPUT parameter if it has not been specified on the DD statement. This parameter must not be specified if the TCIN parameter is specified.

TCIN=*tcin***(***member***)**

PDS and member name of the copybook or template that describes the record structure of your input data.

Note: If you specify a template for DSEB and DSU, it is ignored, except for calls to the external REXX function PRINT specifying TABL or SNGL format.

OFFSETIN

The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 fields within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname

The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

MEMLIST

Allows you to specify a list of member names.

member_n

The name of the member to be processed. Generic name masks are allowed.

Copybook processing

If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG

Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO

Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL

Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM

Invoke the HLASM compiler to create a template from the copybook.

COBOL options

The following options are used to compile a COBOL copybook into a template:

DBCS=YES

Use the DBCS compiler option.

DBCS=NO

Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO

Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES

Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO

Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES

Use the COBOL compile option ARITH(EXTEND).

MIXED = NO

Field names stored in the template in uppercase.

MIXED = YES

Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of "From" and "To" pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to

be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

```
//DSU JOB (acct),'name'  PDS Member Update
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//FMNTSPRT DD SYSOUT=*
//JCLPDS   DD DSN=FMNUSER.FMOS390.JCL,DISP=SHR
//SYSIN    DD *
$$FILEM DSU INPUT=JCLPDS,MEMBER=*,PROC=*
/* Translate all records to uppercase */
Upper outrec
Return
/+
$$FILEM EOJ
/*
```

```
//DSU JOB (acct),'name'  Fix post code
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//FMNTSPRT DD SYSOUT=*
//SYSIN    DD *
$$FILEM DSU DSNIN=FMNUSER.FMOS390.TRANRECS,
$$FILEM     PROC=*
/* Locate name and address record for James  */
/* Browne and change postcode, stored in     */
/* packed decimal, from 6011 to 6194         */
If Substr(inrec,1,1) == 'A' &
   Substr(inrec,32,5) == 'James' &
   Substr(inrec,57,6) == 'Browne' then
   outrec = Change(outrec,'06011F'x,'06194F'x,1,125,3)
Return
/+
$$FILEM EOJ
/*
```

DSV (Data Set View)

Purpose

The DSV function displays the Data Set View or View Entry panel, in online mode. This function can be invoked from a REXX program or a TSO clist or entered on an ISPF Command line, but **cannot** be used in batch jobs as it is an interactive function.

Usage notes

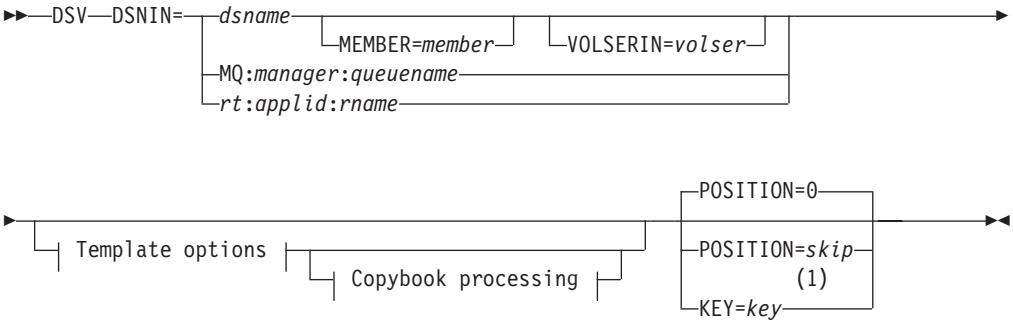
When invoked from an ISPF Command line without a parameter, the Data Set View Entry panel is displayed in interactive mode and you can specify a valid data set for browsing. If you include a valid data set name as a parameter, then you bypass the Entry panel, and go directly to viewing the specified data set.

When invoked from a REXX procedure or TSO list, you must supply the data set name.

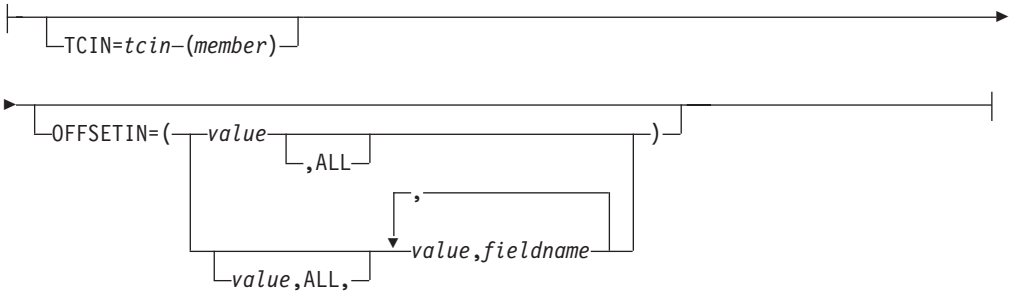
Related functions

DSB Display Data Set Browse or Browse Entry panel
DSE Display Data Set Edit or Edit Entry panel
DSEB Update disk data set records
DSU Update disk data set records

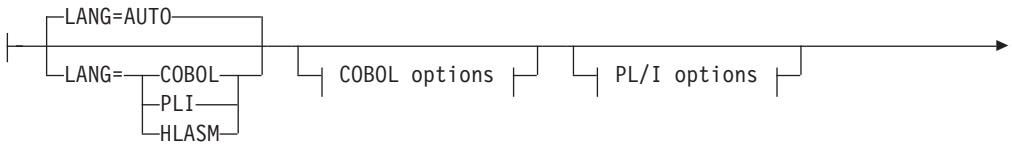
Syntax



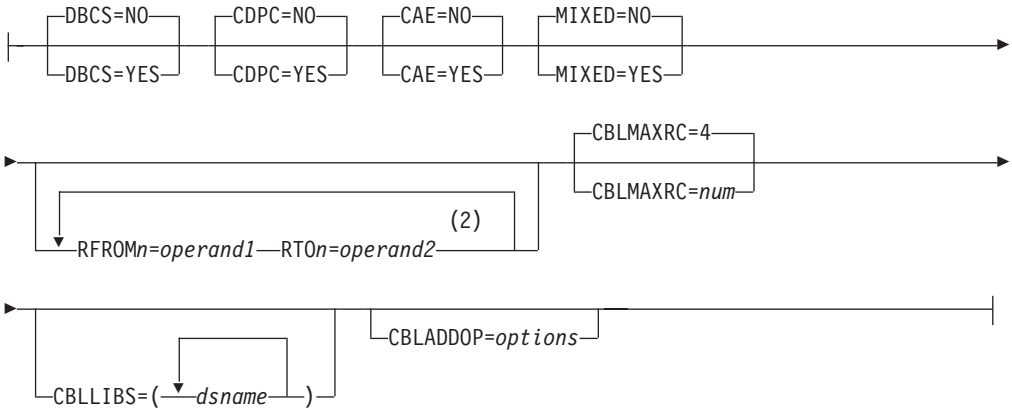
Template options:



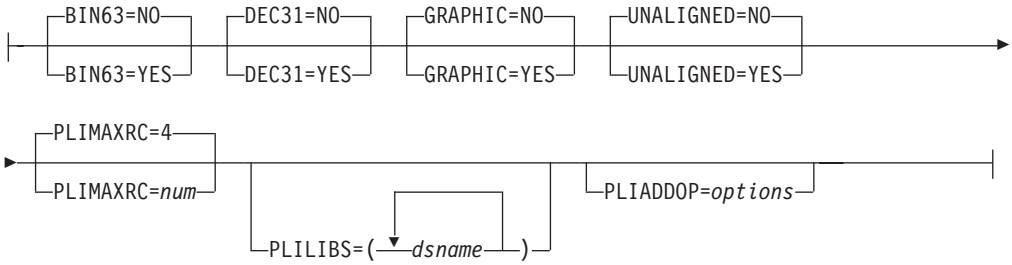
Copybook processing:



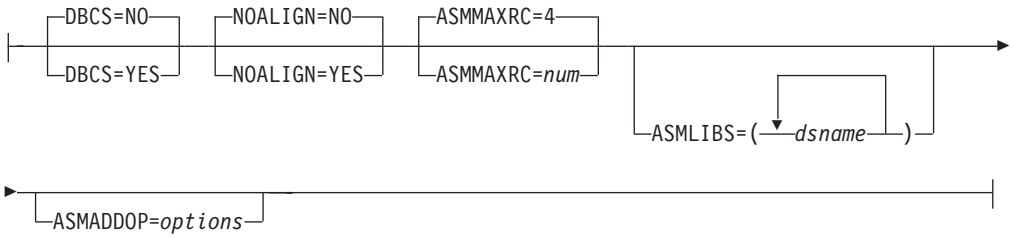
COBOL options:



PL/I options:



HLASM options:



Notes:

1 VSAM only.

2 *n*= 1 to 5.

DSNIN=*dsname*

Defines the name of the data set to be viewed. If specified, any DD statement provided are not used. The name may include a member name

in parenthesis. If the member is specified here, the associated Member parameter must be empty. Must be specified when $DSV is invoked from a REXX procedure or TSO clist.

Can be omitted when DSV is entered on a command line. If specified, you bypass the Entry panel, and go directly to the Data Set View panel. When omitted, the Data Set View Entry panel is displayed in interactive mode and you can specify a valid data set for browsing.

You can further describe this data set, as follows:

MEMBER=*member*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the data set is a PDS, you must specify this parameter.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%`, all members in the PDS whose name is four characters in length are processed.

MEMBER=*member* is ignored if the data set is not a PDS.

VOLSERIN=*volser*
Volume serial number for a non-cataloged data set.

MQ:*manager***:***queuename*
You can specify a MQ queue in place of a data set name, where:

manager
The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from.

queuename
The queue to be used. If you specify a generic name, File Manager displays a list of matching queues to select from.

*rt***:***applid***:***rname*
You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

POSITION=*skip*
Number of logical records to be skipped from the beginning of the data set. The default is 0.

KEY=*key* (VSAM only)
A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, browsing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

TCIN=*tcin*(*member*)
PDS and member name of the copybook or template that describes the record structure of your data set.

OFFSETIN
The length of the 01 field in the template and the start locations of the fields within that 01 field are adjusted by the value provided.

value The offset value, which must be in the range -32760 to 32760, to be applied to the corresponding field identifier. If no field identifier is supplied and ALL is not used, the value is applied to the first Level 01 field in the template.

ALL Where the template contains multiple record structures, this keyword applies the corresponding *value* to all Level 01 within the template.

Note: You can specify a value for ALL and then override this value for individual layouts by providing subsequent *value* and *fieldname* combinations.

fieldname
The name of the Level 01 field to which *value* is to be applied. The default is the first Level 01 field in the template.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

For more information on using this function, see:

- "Invoking File Manager panels from outside File Manager" on page 13
- Chapter 12, "Introduction to programming with File Manager functions," on page 397
- "Invoking File Manager panels from REXX procedures or TSO clists" on page 400

```
FILEMGR "$DSV DSNIN='FMNUSER.TEST.KSDS1',KEY=000100"
FILEMGR DSV
```

DSX (Data Set Extents)

Purpose

Display the extents of a data set.

Usage notes

Use this function to display the "begin" and "end" of the extents of a data set.

Use the information for all basic disk functions which require the location of a disk extent.

For multivolume data sets, only the extents on the selected volume are shown.

Absolute disk extents and disk extents relative to the beginning of the data set are displayed.

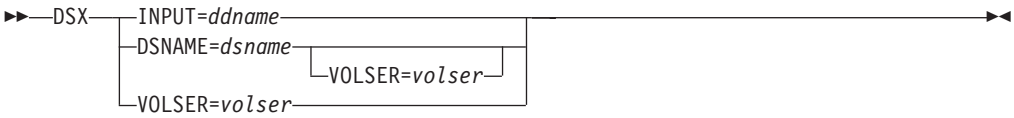
Options

None.

Related functions

DSU Update disk data set records
DVT Print VTOC entries

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

dsname

Name of a disk data set.

volser Volume serial number for a non-cataloged data set.

```
//DP  JOB (acct),'name'  Data Set Extents
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//DISK     DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN    DD *
$$FILEM DSX INPUT=DISK
$$FILEM EOJ
/*
```

DVT (Display VTOC)

Purpose

Display or print a list of the data sets on a disk volume obtained from the disk VTOC.

Usage notes

Use this function to display or print information about data sets and free space on a disk volume. The information is obtained from the disk VTOC. In the list, you can scroll forward and backward, select the alternate display, and locate specific data set names. You can print this list.

You can sort the information by any one of several criteria (see following description for the DVT function variable, *sortdsnby*). The listing also shows free extents on the disk.

Alternatively, you can request that only a VTOC summary, similar to that shown in Figure 85 on page 327, is produced. You can sort the summary list by any one of several criteria (see following description for the DVT function variable, *sortvolby*).

You can also request for a combined data set and summary list report to be produced.

Note: DVT shows "never to expire" expiration dates of 99/999 and 99/366 as 9999.999 and 9999.366, respectively. An expiration date that was explicitly specified as 99/365 is also considered "never to expire", and is shown as 9999.365. However, an expiration date of 99/365 that was calculated from a specified retention period (current date + number of days = 99/365) *will* expire, and is shown as 1999.365. If you sort the list by expiration date, values beginning with 9999 appear first.

Options

Use a fully-qualified or generic file ID, or search patterns to limit the output. The list can be sorted by entry name, creation date, expiration date, or disk extent.

Return codes

The default return codes from the DVT function have the following modified meanings:

4 No data sets found

4 No data set matches pattern on selected volumes

4 No volume matches pattern

16 Volume not mounted

16 Insufficient storage available

16 DVT abended

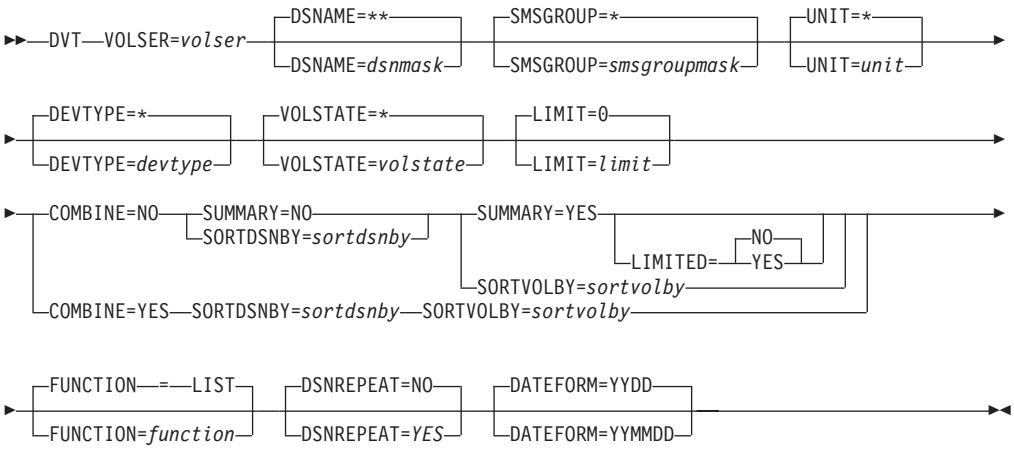
16 Other serious error that stops processing occurred (for example an input or output error)

Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Related function

DSX Display the extents of a data set

Syntax



volser Volume serial number of the disk. Can include a "volume mask" with a percent sign (%) representing exactly one character, or an asterisk (*) representing any number of characters (or no characters). Only volumes matching the mask participate in processing.

dsnmask

A generic data set name, or search pattern. Within the data set name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers. The default is **.

If you also specify a "volume mask" in **volser**, only volumes having data sets matching the dsnmask participate in the processing.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes. For example, the default of ** lists all entries in batch mode; in any other mode, it lists all entries that have the user's TSO prefix as high-level qualifier.

smsgroupmask

A generic SMS storage group name, or search pattern. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters).

unit A generic device address, or search pattern. Within the address, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters).

devtype

A generic device type name, or search pattern. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters).

volstate

A generic volume status name, or search pattern. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters).

The following full names are supported:
- PRIVATE (also includes PRIV/RSDNT)
- RESIDENT
- PUBLIC
- STORAGE
- ALL

limit The maximum number of data sets to be selected for processing. The default (0) indicates that there is no limit on the number of data sets selected.

function
The command that you want to execute:
LIST Display the list (the default)
PRINT
Print the list

SUMMARY
Determines whether or not the VTOC list is produced in full or as a summary.

NO (Default) The full list is produced.

YES A summary is produced and the SORTBY parameter is ignored. The list is sorted by device number.

LIMITED
Determines where the volume summary list (SUMMARY=YES) contains detailed (NO) or limited (YES) data about the number of datasets located on the volumes processed. Specifying LIMITED=YES provides faster response times.

NO Detailed dataset information included in the volume summary list.

YES No dataset counts provided in the volume summary list.

sortdsnby
Determines the sequence in which the VTOC data set list appears:
NAME
Sorted by data set name
VOLSER
Sorted by volume serial
EXTENT
Sorted by extent
SIZE Sorted by data set size
TYPE Sorted by DSORG
RECFM
Sorted by RECFM
LRECL
Sorted by LRECL
BLKSIZE
Sorted by BLKSIZE
DATE Sorted by creation date
EDATE
Sorted by expiration date
RDATE
Sorted by last referred date

sortvolby
Determines the sequence in which the VTOC volume list appears:

VOLSER
Sorted by volume serial

UNIT Sorted by device address (unit)

DTYPE
Sorted by device type

SMSSG
Sorted by SMS storage group

TTTRK
Sorted by disk capacity

USTRK
Sorted by tracks used

USPTRK
Sorted by utilization in %

TRKF Sorted by tracks available

TDSN Sorted by number of data sets

TVSDSN
Sorted by number of VSAM data sets

TNVSDSN
Sorted by number of non-VSAM data sets

SVTOC
Sorted by VTOC size

UPVTOC
Sorted by VTOC utilization in %

INDX Sorted by VTOC indexing

DSCBS
Sorted by VTOC free DSCBs

FCYL Sorted by cylinders available

MFCYL
Sorted by maximum free space in cylinders

FTRK Sorted by tracks available

MFTRK
Sorted by maximum free space in tracks

COMBINE
Determines whether a combined data set and volume summary list report is produced.

NO Does not produces a combined data set and volume summary list report.

YES Produces a combined data set and volume summary list report.

DSNREPEAT
Determines whether a combined data set and volume summary list report is produced.

NO Suppresses the data set name on the second and subsequent extents.

YES The data set name appears for each extent.

DATEFORM
Determines the display format for the date fields.

YYDD Use the date format YYYY.DDD.

YYMMDD
Use the date format YY/MM/DD.

```
//DVT JOB (acct),'name'  Display VTOC
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM DVT VOLSER=SCPMV5,DSNAME=**,SORTBY=DATE
$$FILEM EOJ
/*
```

EOJ (End of Job)

Purpose
Exit File Manager.

No parameters.

ERT (Erase Tape)

Purpose
Erase a tape from the current position until the end of the tape.

Usage notes
This function writes two tape marks at the current tape position, erases the remainder of the tape, then rewinds the tape.

ERT can be used only with tape units that support the data security erase I/O command code, such as the IBM 3400 series.

Options
None.

Related functions
REW Rewind a tape to its load point
RUN Rewind a tape and unload it
WTM Write one or more tape marks

Syntax

ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be

specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

EVC (Exported Stacked Volume Copy)

Purpose

Copy a logical volume from an Exported Stacked Volume to a physical volume.

Usage notes

If either the input volume or the output volume are not at their load point, this function rewinds the volumes.

The input volume is verified to ensure that it was created by the VTS export function. If valid, then the Table of Contents located at the end of the Exported Stacked Volume volume is read. The Exported Stacked Volume is then rewound again before the copy process starts.

The requested logical volumes are then copied to the physical volumes. The copy sequence is the same as the sequence of the logical volumes on the Exported Stacked Volume. The requested list of logical volumes is internally sorted into the same sequence as the Exported Stacked Volume Table of Contents to enable all copies to be completed without the need to rewind the input volume during the copy process.

Specify a DDNAME for the tape. If the DDNAME is not allocated, you are asked for allocation information.

You can copy up to 5 logical volumes in a single step.

If you do not know the VOLSER and first data set name of the tape, specify BLP during allocation.

For more information about using this function, see "ESV options" on page 343.

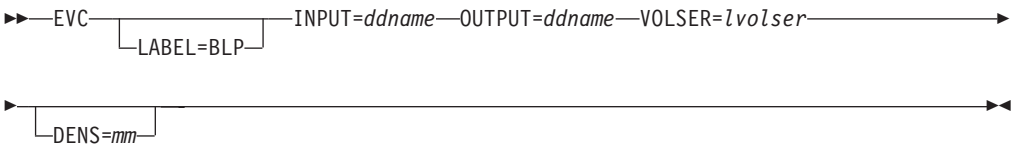
Options

None.

Related function

EVL Print a table of contents list for an Exported Stacked Volume

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

There must be one to one correspondence between the logical volumes requested (*lvolser*) and the physical output volumes specified in the VOL= parameter of the DD statement specified by *ddname* in the OUTPUT parameter.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

lvolser

Logical volume numbers of the volumes to be copied. Use VOL1 for a standard labeled volume. Specify up to five logical volume numbers in the form (*volser1, volser2, ...*).

The logical volume numbers must be unique, duplicates are not permitted (that is, one logical volume cannot be specified more than once). Each logical volume is copied to a single physical output volume (specified in *tapevol*).

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

```
//EVC JOB (acct),'name'  Exported Stacked Volume Copy
//* Copy 1 logical volume to 1 specified volume
//DTOEVC   EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPE1    DD UNIT=561,VOL=SER=ESV001,
//            DISP=SHR,LABEL=(,BLP)
//TAPE2    DD UNIT=562,VOL=SER=NEWVOL,
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM EVC INPUT=TAPE1,OUTPUT=TAPE2,VOLSER=LVOL03
$$FILEM EOJ
/*
```

```
File Manager
$$FILEM EVC INPUT=TAPE1,OUTPUT=TAPE2,VOLSER=LVOL03
Logical volume LVTS03 being copied to physical volume NEWVOL
Standard Label File Number    1  DSN='LVOL03.TEST '
Copy of logical volume LVOL03 to physical volume NEWVOL complete
EVC completed
```

```
//EVC JOB (acct),'name'  Exported Stacked Volume Copy
//* Copy 2 logical volumes to 2 specified volumes
//DTOEVC   EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPE1    DD UNIT=561,VOL=SER=ESV001,
//            DISP=SHR,LABEL=(,BLP)
//TAPE2    DD UNIT=562,VOL=SER=(NEWVL1,NEWVL2),
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM EVC INPUT=TAPE1,OUTPUT=TAPE2,
$$FILEM     VOLSER=(LVOL03,LVOL06)
$$FILEM EOJ
/*
```

```
//EVC JOB (acct),'name'  Exported Stacked Volume Copy
//* Copy 5 logical volumes to 5 scratch volumes
//DTOEVC   EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPE1    DD UNIT=561,VOL=SER=ESV001,
//            DISP=SHR,LABEL=(,BLP)
//TAPE2    DD UNIT=562,DISP=(NEW,PASS),LABEL=(,SL)
//SYSIN    DD *
$$FILEM EVC INPUT=TAPE1,OUTPUT=TAPE2,
$$FILEM     VOLSER=(LVOL03,LVOL06,LVOL01,LVOL99,LVOL10)
$$FILEM EOJ
/*
```

EVL (Exported Stacked Volume List)

Purpose

List the contents of an Exported Stacked Volume.

Usage notes

File Manager first verifies that the volume loaded was created by the VTS export function. File Manager forward spaces the volume to the Table of Contents and then prints a listing of the Table of Contents. The sequence printed is the same as the sequence of the logical volumes on the Exported Stacked Volume.

You can choose a short listing using the SHORT listing type, which prints only the logical volumes on the Exported Stacked Volume, or you can obtain a more detailed list of the Table of Contents by using the LONG listing type. This listing shows standard label header and trailer information and the location of tape marks.

Specify a DDNAME for the tape. If the DDNAME is not allocated, you are asked for allocation information.

If you do not know the VOLSER and first data set name of the tape, specify BLP during allocation.

For more information about using this function, see "ESV options" on page 343.

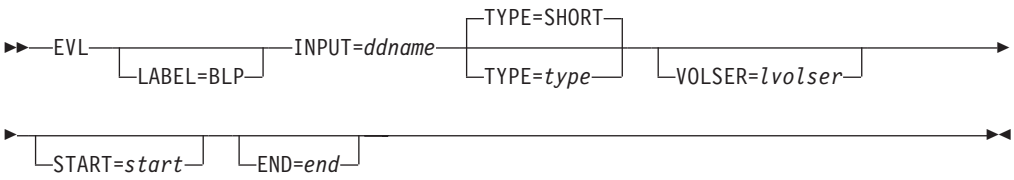
Options

None.

Related function

EVC Copy a logical volume from an Exported Stacked Volume

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

end The relative number of the logical volume on the input Exported Stacked Volume where the listing is to stop. For example, END=9 causes the listing of logical volumes to stop after volume 9 has been listed. If omitted, or greater than the number of logical volumes on the input Exported Stacked Volume, then the value defaults to the maximum number of volumes on the input Exported Stacked Volume. Ignored if TYPE=SHORT, or VOLSER has been specified.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

lvolser
Logical volume numbers of the volumes to be listed. Use VOL1 for a standard labeled volume. Specify up to five logical volume numbers in the form (*volser1*, *volser2*, ...). If omitted, and START, END parameters are not used, then all logical volumes on the input Exported Stacked Volume are listed. Ignored if TYPE=SHORT.

start The relative number of the logical volume on the input Exported Stacked Volume where the listing is to start. For example, START=5 causes the first 4 logical volumes to be skipped and the listing starts with logical volume 5. If omitted, the value defaults to 1. If START is greater than END, then no logical volumes is listed. Ignored if TYPE=SHORT, or VOLSER has been specified.

type Specify one of the following:

SHORT
Print only the volsers of the logical volumes on the input Exported Stacked Volume (the default).

LONG
Print detailed listing of the format of the logical volumes on the input Exported Stacked Volume. Parameters VOLSER or START, END can be used to qualify the logical volumes listed.

```
//EVL JOB (acct),'name'  Exported Stacked Volume List
//DTOEVL   EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=F07M3N80,VOL=SER=ESV994,
//             DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM EVL INPUT=TAPE,TYPE=LONG,VOLSER=(EJ0020,EJ1019,EJ1029,EJ1035)
$$FILEM EOJ
/*
```

```
File Manager
$$FILEM EVL INPUT=TAPE,TYPE=LONG,VOLSER=(EJ0020,EJ1019,EJ1029,EJ1035)
Exported Stacked Volume Table of Contents for Volume ESV994
          VOLSER   SEQ NO    Logical Volume Layout
          EJ1019      3      VOL1EJ1019
                             HDR1PE.MNT06.TAB3510O
                             ----- TAPE MARK -----
                                   Data File
                             ----- TAPE MARK -----
                             EOF1PE.MNT06.TAB3510O
                             ----- TAPE MARK -----
                             ----- TAPE MARK -----
                             ===== End of Volume =====
          EJ1035      4      VOL1EJ1035
                             HDR1PE.MNT12.TAB3510O
                             ----- TAPE MARK -----
                                   Data File
                             ----- TAPE MARK -----
                             EOF1PE.MNT12.TAB3510O
                             ----- TAPE MARK -----
                             ----- TAPE MARK -----
                             ===== End of Volume =====
          EJ1029      7      VOL1EJ1029
                             HDR1PE.MNT08.TAB3510O
                             ----- TAPE MARK -----
                                   Data File
                             ----- TAPE MARK -----
                             EOF1PE.MNT08.TAB3510O
                             ----- TAPE MARK -----
                             ----- TAPE MARK -----
                             ===== End of Volume =====
```

```
          EJ0020     10      VOL1EJ0020
                             HDR1.BACKTAPE.DATASET
                             ----- TAPE MARK -----
                                   Data File
                             ----- TAPE MARK -----
                             EOF1.BACKTAPE.DATASET
                             ----- TAPE MARK -----
                             ----- TAPE MARK -----
                             ===== End of Volume =====
EVL completed
$$FILEM EOJ
```

FCH (Find/Change)

Purpose

The FCH function allows you to:

- Search for, and optionally change, strings in a PDS, VSAM data set, sequential data set, MQ queue, or CICS resource.
- Search for strings in HFS files.

Usage notes

You can select the records to be processed using:

- Member name selection criteria
- Date created selection criteria
- Date last modified selection criteria
- User ID selection criteria

You can either specify a REXX procedure with the *proc* parameter, or enter a FIND or CHANGE command in the Command line. For information about the primary commands, see "Finding and changing data in multiple PDS members" on page 270. The LOCATE primary command is ignored in batch jobs. When working with compressed non-VSAM extended format data sets (compressed PSE data sets), the CHANGE command is not supported, however, the FIND command can be used.

Multiple command processing

There is no limit on the number of FIND or CHANGE commands that can be processed in one pass of the file, but each FIND or CHANGE command must start on a new line. Be careful when using overlapping change commands such as `C cat dog` and `C catapult crossbow`. A string is only matched against the first command with a matching search argument. Therefore, you must place the longer change first. For example, if you specify the following change commands:

```
C Cat Dog
C Catapult Crossbow
```

The second command would never get processed. Reversing the command order would ensure any occurrences of "Catapult" were changed.

After a change is made, FCH processing continues for the same data record. The point where processing resumes is immediately following the most recently changed string. As a result, changes are not recursive, so that `C cat cow` and `C cow dog` does not change "cat" to "dog", unless separate runs are done. Furthermore, if the search argument is found but the change fails, subsequent FIND or CHANGE commands that match that string are not done.

Performance tips

- When you use FCH to update members of a PDS(E):

  One FCH default is STATS=ON, which causes the ISPF statistics for each changed member to be updated. This can significantly increase I/O (EXCP) and CPU utilization. To improve performance, consider using STATS=OFF.
- Using JCL processing (JCL=YES) is more CPU intensive than JCL=NO. Only use JCL=YES if necessary.
- You can improve concurrent read access by other users or jobs to the target data set when there is a user PROC, by using the NOUPDATE=YES option when the PROC will not be performing any updates.

Options

When you specify the PROC option, you are supplying a REXX procedure. For more information, see the *proc* parameter below.

Return codes

The default return codes from the FCH function have the following modified meanings:

1 One or more FIND or CHANGE commands successful but one or more FIND or CHANGE commands unsuccessful because no strings found.

2 One or more strings found but one or more CHANGE commands could not be performed (no space available or invalid key change).

4 No FIND or CHANGE command successful because no strings found (no matches).

4 No FIND or CHANGE command successful because no members to process.

4 No FIND or CHANGE command successful because input empty.

8 Bad FIND/CHANGE command(s) supplied.

8 Too many FIND/CHANGE commands supplied.

8 REXX error encountered.

8 Job step interrupted/cancelled.

16 Program Object not supported (but specified).

16 Data set in use.

16 Member in use.

16 Data set/member open error.

16 Data set not found / allocation error.

16 Insufficient storage available.

16 Input data appears ISPF packed but is not valid.

16 FCH abended.

16 Other serious error that stops processing occurred (for example an input/output error).

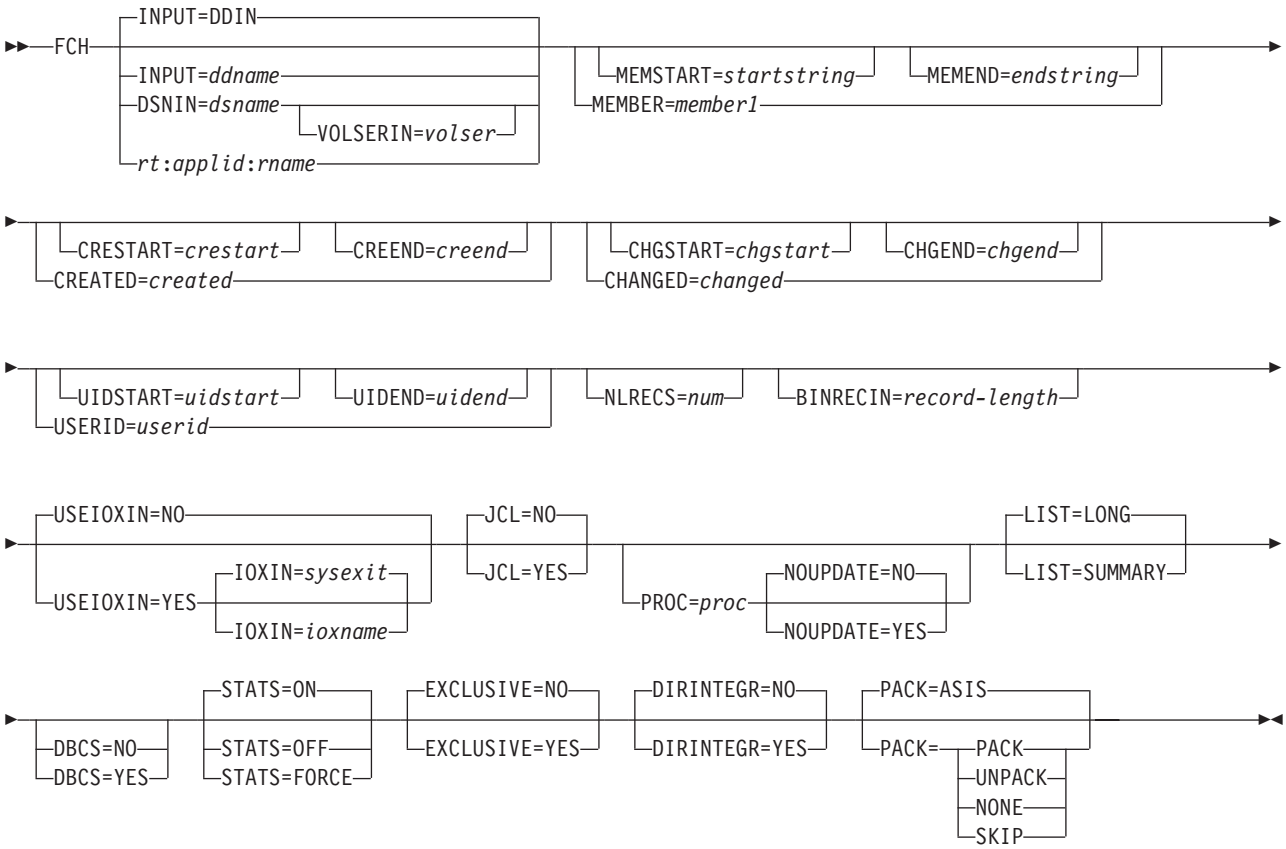
Note: Return codes can be customized during installation. If you receive return codes that do not match those listed above, your site might have customized the return codes in place for this function. File Manager may

also issue the 999 abend, if the return code in batch is equal to or greater than the ABENDCC value. Please contact your File Manager systems administrator for details.

Related functions

DSEB Edit a data set via batch job processing
TRS Locate data within a tape file

Syntax



INPUT=*ddname*

Defines a reference to a DD or TSO ALLOC statement for the input data set, or HFS file. The default is `DDIN`.

DSNIN=*dsname*

Defines the name of the input data set, or an absolute path to a HFS file (directory). . If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. You can further describe this data set, as follows:

VOLSERIN=*volser*

Volume serial number for a non-cataloged data set.

An absolute path to a HFS file (directory) must be enclosed in apostrophes. If it does not fit on one line, it can be split into several lines.

rt:*applid*:*rname*

You can specify a CICS resource in place of a data set name, where:

rt Resource type. Valid values are:
FI For a CICS file.
TD For a Transient Data Queue.
TS For a Temporary Storage Queue.

applid The VTAM applid of the CICS system.

rname The name of the resource.

MEMBER=*member1*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS. If the input data set is a PDS(E), you may specify this parameter, or a member name in the DD statement for *ddname*, or specify a range of member names with the MEMSTART and MEMEND keywords.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member1 is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the copy. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the copy. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member1* parameter of the MEMBER keyword.

CREATED=*created*
The date on which a member was created, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of creation dates with the CRESTART and CREEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

created is ignored if the data set is not a PDS.

CRESTART=*crestart*
The start of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If CRESTART is specified but CREEND is omitted, all members of the PDS(E) from the *crestart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *crestart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CREEND=*creend*
The end of a range of creation dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *creend* defaults to the right as follows:
DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

CHANGED=*changed*
The date on which a member was last modified, in YYYY/MM/DD format.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of modification dates with the CHGSTART and CHGEND keywords.

You can specify an asterisk (*) as the last character to indicate a range of dates or a percent sign (%) in place of a single character to indicate a selection of dates.

changed is ignored if the data set is not a PDS.

CHGSTART=*chgstart*
The start of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If CHGSTART is specified but CHGEND is omitted, all members of the PDS(E) from the *chgstart* value onwards are included.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgstart* defaults to the right as follows:
DD = 01
MM = 01
YYYY = 0000

No other wildcarding is allowed.

CHGEND=*chgend*
The end of a range of modification dates in YYYY/MM/DD format to be included in the copy.

If omitted, or you do not enter a full date, or you specify an asterisk (*) as the last character, the unspecified portion of *chgend* defaults to the right as follows:

DD = 31
MM = 12
YYYY = 9999

No other wildcarding is allowed.

USERID=*userid*
The TSO user ID by which the member was last updated.

If the input data set is a PDS(E), you may specify this parameter, or specify a range of user IDs with the UIDSTART and UIDEND keywords.

You can enter a generic user ID by using asterisks and percent signs.

userid is ignored if the data set is not a PDS.

UIDSTART=*uidstart*
The start of a range of user IDs to be included in the copy.

If UIDSTART is specified but UIDEND is omitted, all members of the PDS(E) from the *uidstart* value onwards are included.

If omitted, or you do not enter a full 7-character user ID, or you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidstart* to the right with low values (X'00').

UIDEND=*uidend*
The end of a range of user IDs to be included in the copy.

If you omit this field, it defaults to high values (X'FF').

If you specify less than 7 characters (without an asterisk as the last character), File Manager pads *uidstart* to the right with low values (X'00'). If you specify an asterisk (*) as the last character, File Manager replaces the asterisk and pads the unspecified portion of *uidend* with high values (X'FF').

NLRECS=*num*
Specifies the number of records to be processed in each data set or member.

BINRECIN=*record-length*
Specifies the record length used for processing a HFS file. Valid range: 1 to 32760. The file is processed in binary mode (fixed-length records derived from the file, delimiters not distinguished). If you do not specify this parameter, the file is processed in text mode (variable-length records, boundaries determined by delimiters).

USEIOXIN
Specifies whether to invoke a user I/O exit, to process the input data set.

NO Default. Do not invoke a user I/O exit.

YES Invoke a user I/O exit to process the input data set. This option is only available if the person who did the site customization for File Manager allowed user I/O exits on a site-wide basis.

IOXIN
Specifies the name of the user I/O exit used for the input data set. There are no restrictions on the programming language that you can use to write an exit. The exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOXIN=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided in the installation customization options. If USEIOXIN has been set to YES and no installation default has been provided, you must specify IOXIN=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname
The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

JCL=NO
Treat the data set as a non-JCL data set.

JCL=YES
The data set contains JCL and the JCL syntax is to be preserved.

The columns searched are set to 3 through 71, unless the statement is not a JCL statement. A statement is considered to be a JCL statement if it begins with the strings "/*" or "//". If the statement does not begin with either of these strings, it is not considered to be a JCL statement in which case any column range specified on the FIND (or CHANGE, respectively) command or preset using the BOUNDS command is honored. If no column range has been specified, the full record is searched.

If not successful at maintaining the number and size of records, File Manager attempts to rewrite the file:

- More errors are possible in this case. For example, a PDS(E) may run out of room.
- If a logical line is changed and requires more physical records, the file is rewritten. The data in columns 73–record length for new physical records is copied from the last related original physical record. Data past column 72 is treated as non-changeable sequence numbers or comments.

PROC=*proc*
Member name of a REXX procedure that you want to use to process each record, or an asterisk (*) to indicate the procedure is inline. If you specify a member name, you must define an FMNEXEC ddname that identifies the PDS containing the member. If you specify *, the procedure is read from SYSIN immediately following the control statement for the current function. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

Whether or not a record appears in the FCH report is determined by the return code from the REXX procedure for that record. (If no PROC statement is specified, one is assumed at the end of the $$FILEM control statements.) If the REXX procedure ends with a RETURN DROP statement, then the current record is considered to be “not selected” (not one of the records you wanted to find), and does not appear in the FCH report. If the REXX procedure ends normally, or with an explicit RETURN (without the DROP keyword), then the current record is considered to be “selected”, and is included in the FCH report. Records that have been selected without being changed by the REXX procedure are marked in the FCH report by an “s” suffix attached to their record number, while records that have been selected and changed are marked by a “c”.

In a REXX procedure for FCH, explicitly code a RETURN statement when you identify a record that you want to select. To ensure that other records are not selected, on the last line of the REXX procedure, code a RETURN DROP statement.

For more information about using REXX procedures to process records, see Chapter 13, "Enhancing File Manager processing," on page 405.

NOUPDATE
Allows you to specify that you intend no updates to the FCH data set while executing the utility. This option is valid only when a REXX procedure has been specified and is ignored otherwise.

NO Updates to the data are honored.

YES Forces the allocation of the data set as input only. All updates to the data are ignored.

LIST=LONG
Default. This prints each record where the string was found as well as a summary report.

LIST=SUMMARY
This produces a summary report only.

DBCS=YES
(Default for LANGUAGE=JAPANESE). This processes and preserves DBCS shiftin and shiftout characters within the data records.

DBCS=NO
(Default for LANGUAGE=ENGLISH). This ignores DBCS shiftin and shiftout characters within the data records.

STATS=ON
Default. This updates the ISPF statistics (if already present) when a PDS or PDSE member has been changed.

STATS=OFF
The ISPF statistics is not updated when a PDS or PDSE member has been changed.

STATS=FORCE
The ISPF statistics that exist for members being processed are always updated and statistics for a member that previously did not have statistics are created.

EXCLUSIVE=NO

Note: This option is supported for backward compatibility only.

Use the new DIRINTEGR option.
Default. The data set is allocated with DISP=SHR, so that other users can obtain concurrent access to a PDS or PDSE during execution of FCH.

EXCLUSIVE=YES
Forces an override of the PDS(E) member processing method which allows for safe concurrent updates by other users. This option has significant performance impact. When set to YES, the member processing is performed much faster but may be affected by PDS(E) directory updates, possibly causing I/O errors if the data set is concurrently updated. This option overrides the processing method selected by File Manager (EXCLUSIVE=NO, default, unless the input data set has been allocated

OLD by the user), which always assumes concurrent safe processing when the data set is allocated to multiple users.

DIRINTEGR
Specifies whether to invoke a user I/O exit to process the input data set.

NO Default. File Manager uses a faster PDS(E) directory processing method. This may cause I/O errors when multiple users are concurrently updating the directory of the data set being processed.

YES File Manager uses safer, but slower, PDS(E) directory processing method. This method allows for safe concurrent updates of the PDS(E) directory by multiple users.

PACK Determines if File Manager should detect if the input data is in ISPF packed format and specifies if the output data is to be written in ISPF packed format. This keyword is ignored when processing VSAM data sets. When an I/O exit has been specified for either the input or output data set (or both), the only valid option is PACK=NONE.

ASIS Instructs File Manager to write the output in ISPF Packed format only if the input is in ISPF packed format.

PACK Instructs File Manager to write the output in ISPF packed format regardless of the input format.

UNPACK
Instructs File Manager to write the output without ISPF packing, regardless of the input format.

NONE
Instructs File Manager not to determine if the input data set is in ISPF packed format and writes the output records as they are read from the input data set (after any enhanced processing).

SKIP Instructs File Manager to determine if the input data set is in ISPF packed format and if so, to skip the find/change processing.

See "Finding and changing data in multiple PDS members" on page 270 for details of the FCH commands.

Note: It is not possible to use multiple FINDNOT commands in the input stream for batch processing. Similarly, it is not possible to combine FINDNOT commands with FIND and/or CHANGE commands in the batch input stream.

```
//FMUSRFCH JOB  (@TS2,MVS6),'FMNUSER',NOTIFY=FMNUSER,
//  CLASS=A,MSGLEVEL=(1,1),MSGCLASS=H
//FMNBAT   EXEC PGM=FILEMGR
//STEPLIB  DD  DSN=FMNUSER.FMN110.TSTLOAD,DISP=SHR
//         DD  DSN=FMN.V1R1M0.TSTLOAD,DISP=SHR
//         DD  DSN=FMN.V1R1M0.SFMNMOD1,DISP=SHR
//         DD  DSN=FMN.IGYV1R20.SIGYCOMP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FMNEXEC  DD DSN=FMN.EXEC,DISP=SHR
//FMNTSPRT DD SYSOUT=*
//FMNIN    DD DSN=FMNUSER.JCL.TESTING,DISP=SHR
//SYSTERM  DD SYSOUT=*
//SYSIN    DD *
$$FILEM FCH ,
$$FILEM INPUT=FMNIN,MEMBER=J*
F 'rights reserved'
C 'Copyright 2001-2002' ,
'Copyright 2001-2003' 1 71
```

```
C 'Alpha Company Ltd' ,
'Alpha Beta Company Ltd' 1 71
C 'Beta Company Ltd' ,
'Alpha Beta Company Ltd' 1 71
/+
/*
```

FMT (Set DBCS Format)

Purpose

Specify the format of records that contain double-byte character set (DBCS) data.

Usage notes

File Manager considers data as being contained in one or more fields in the input record. This function lets you define where each type of field is located within the record. You can define up to 32 fields in ascending order. They must not overlap, but they need not be contiguous. Specify for each field: start column, end column, type. An asterisk (*) for end column indicates end-of-record. For type, you can specify EB (for EBCDIC format), DB (for DBCS format), or MI (for mixed format).

For all areas you do not define, File Manager creates an EBCDIC field definition.

If you use the FMT function within a batch job, your format definition affects only the current batch job. In any other case, your format definition is saved in the user profile and is available for further File Manager sessions.

The FMT function and the "DBCS capable language" processing option of the SET function work together as follows:

- The FMT function specifies which data within each record is DBCS.
- The "DBCS capable language" option of the SET function causes DBCS data to be printed correctly.

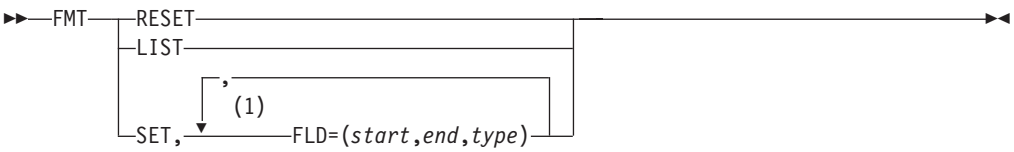
For more information about the SET function, see "SET (Set Processing Options)" on page 1047.

Note: The FMT function has no effect on data printed in TABL or SNGL print format by a print function in advanced-function mode when the language used is a DBCS capable language or when using TSO if the terminal used is DBCS capable. However, when either the language is a non DBCS language or the TSO terminal is not DBCS capable then the DB type can be used to format graphic fields correctly in the output.

Options

RESET specifies that the default (1,*,EB) is to be used. LIST prints a listing of the current FMT setting.

Syntax



Notes:

1 You can specify up to 32 fields using the FLD keyword.

LIST Lists all the field definitions, whether you defined them explicitly or File Manager used the default.

RESET
Resets all field definitions to the default FLD=(1,*,EB).

SET Defines a new set of fields.

start Starting position of a field, in bytes, where 1 represents the start of the record.

end Ending position of the field, in bytes. Specify an asterisk (*) for the end of the record.

type Type of a field:

DB DBCS data. The field length must be an even number of characters. Unformatted graphic fields are not treated as DBCS.

MI Mixed DBCS and EBCDIC data. Data between the shift out character X'0E' and the shift in character X'0F' is treated as DBCS data.

EB EBCDIC data (the default).

The field definitions must be in ascending order, and must not overlap. Any areas that you do not define are considered EBCDIC data.

You can define up to 32 fields.

For example, the command:

```
FMT,SET,FLD=(10,27,MI),FLD=(40,59,DB)
```

gives you these field definitions:

```
FIELD1 (1,9,EB)
FIELD2 (10,27,MI)
FIELD3 (28,39,EB)
FIELD4 (40,59,DB)
FIELD5 (60,*,EB)
```

File Manager prints a mask of the record, with a scale to help you to adjust your definition:

```
EEEEEEEEEMMMMMMMMMMMMMMMMMMEEEEEEEEEEEEEDBDBDBDBDBDBDBDBDBE*
1...5...10....5...20....5...30....5...40....5...50....5...60.
```

The valid double-byte hexadecimal values are X'0000', X'4040', and X'4141' to X'FEFE'. File Manager replaces all other pairs of double-byte characters with X'4040'.

```
//FMT JOB (acct),'name'  Define DBCS Record Fmt
//FILEMGR   EXEC PGM=FILEMGR
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
$$FILEM FMT SET,FLD=(10,27,MI),FLD=(40,59,DB)
$$FILEM EOJ
/*
```

FSF (Forward Space File)

Purpose
Move the tape forward one or more tape files.

Usage notes
A tape file consists of blocks of data delimited by a tape mark. For this

function, a tape label set is also considered a file. The tape stops after the specified number of tape marks has been read. The tape is positioned after the tape mark and a following read or write command reads or overwrites the record following that tape mark.

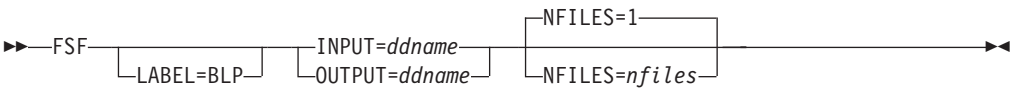
Options

None.

Related functions

BSF Move the tape backward one or more files
BSR Move the tape backward one or more records
FSR Move the tape forward one or more records
REW Rewind a tape to its load point
RUN Rewind a tape and unload it

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nfiles Number of tape files to be spaced forward. The maximum is 99 999; the default is 1. Leading tape marks are also counted.

FSR (Forward Space Record)

Purpose

Move the tape forward one or more tape records or tape marks.

Usage notes

A physical tape record is either a block of data bytes or a tape mark. A tape mark is treated as a special record. For each tape mark crossed, File Manager displays an information message.

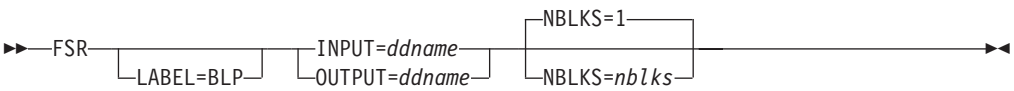
Options

None.

Related functions

BSR Move the tape backward one or more records
BSF Move the tape backward one or more files
FSF Move the tape forward one or more files
REW Rewind a tape to its load point
RUN Rewind a tape and unload it

Syntax



ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nblks Number of tape blocks to be spaced forward. The maximum is 99 999 999; the default is 1. A tape mark is considered a block.

INT (Initialize Tape)

Purpose
Initialize tapes in IBM standard label format, ASCII label format, or NL tape.

Usage notes
Use this function to initialize a labeled or unlabeled tape. File Manager first rewinds the tape. To initialize with labels, File Manager writes a volume label set in either IBM standard format or ANSI type 3 or 4 (ASCII) format. For NL initialization, a single tape mark is written.

Specify a DDNAME for the tape. To check existing labels on the output tape before creating the new label set, specify the appropriate label value on the tape allocation panel. Specifying BLP initializes the tape unconditionally.

After completion of the function, the tape remains positioned after the label set. To make your system aware of the new volume serial number, rewind, unload, and reallocate it with the new volume serial number.

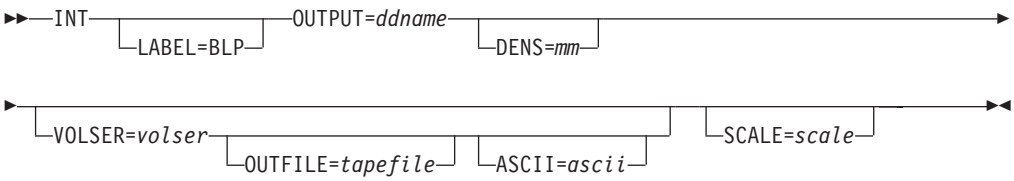
Options
You can initialize a tape with ANSI type 3 or 4 labels to be used with ASCII files. You can also specify a file identifier to be used for the file header label. Omit the volser to initialize an NL tape.

A scaling capacity can be added to the tape provided the function is supported by the hardware. The scaling capacity limits the amount of the tape actually used.

Related function

WTM Write one or more tape marks

Syntax



ascii Specify YES to format the tape with American National Standard labels for use with ASCII files. Specify NO to format the tape with EBCDIC labels. The default depends on the value of the SET TAPELBL parameter.

ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

scale Limits the amount of data that can be stored on the tape to a percentage of its capacity. Allowed values are 0 (100% capacity; no limit applied), 25, 50, or 75. For example, SCALE=25 limits the tape to 25% of its normal capacity. Only valid for 3590 devices; it is not valid for a 3590 running in 3489E emulation mode.

tapefile
Name to be used to write the HDR1 label. The default is to create a HDR1 label with a dummy name for labeled tapes. This parameter is ignored for NL tapes.

volser Volume serial number to be used for initialization. If omitted, an NL tape with no volser is produced.

```
//INT JOB (acct),'name'  Initialize SL Tape
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=561,VOL=SER=FMO001,
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM INT OUTPUT=TAPE,VOLSER=FMO001,
$$FILEM     SCALE=25,OUTFILE=MY.TEST.TAPE
$$FILEM EOJ
/*
```

NOSORT (Disable the use of DFSORT)

Purpose
Disable the use of DFSORT for copy or print processing of data sets.

Usage notes
File Manager may use DFSORT internally for processing to achieve better performance. Under certain conditions (for example, unlike concatenated input data sets), such use may result in DFSORT processing errors and should be disabled.

Syntax

ODL (Object Directory List)

Purpose
Work with a list of OAM objects from the specified collection or optical disk.

Usage notes

You can print the list.

Select the objects by a fully-qualified or a generic object name. Specify the collection name, the volser of an optical volume, or both.

The information listed by ODL includes:

- Object name
- Creation timestamp
- Object size
- Storage location (DASD or Library)
- Volser information (if known)
- Storage class
- Management class
- Expiration date

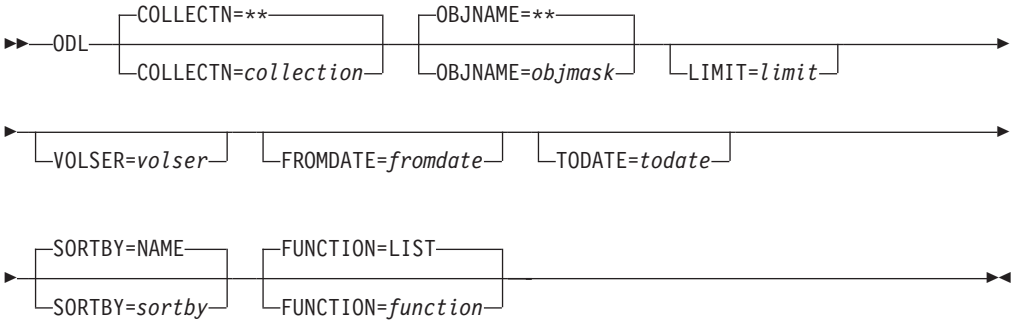
Options

Limit processing to objects residing in DASD storage, on optical storage media, or on a specific optical volume. You can also specify a range for the creation date. The information displayed can be sorted by name, creation date, size or expiration date.

Related functions

OE Erase an OAM object
OO Copy an object to the same or another collection
OP Print an object in character or hexadecimal dump format
OS Backup objects from an OAM database to a data set
OV Backup objects from an OAM database to a VSAM data set

Syntax



collection

A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.

fromdate

The earliest creation date, in the form *yyyymmdd*. By default, all objects are listed regardless of their creation date.

function

The command that you want to execute:

LIST Display the list (the default)

PRINT

Print the list.

limit To limit the list based on where objects are located, specify one of the following:

DASD
Only objects that are stored on DASD

OPTICAL
Only objects that are stored on optical disks

VOLUME
Only objects that are stored on a specified optical disk. You must also specify the volser of the optical disk.

objmask
An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

sortby One of the following:

NAME
Sorts by object name (the default)

DATE Sorts by creation date

EDATE
Sorts by expiration date

SIZE Sorts by object size.

todate The latest creation date, in the form *yyyymmdd*.

volser If you specified VOLUME for *limit*, the volume serial number of the optical disk.

```
//JOB (acct),'name',Optical Volume Contents
//S1       EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM ODL LIMIT=VOLUME,VOLSER=VBKUP6
$$FILEM EOJ
/*
```

OE (Object Erase)

Purpose
Erase an OAM object.

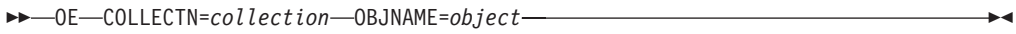
Usage notes
Use this function to remove an object from a specified collection.

Options
None.

Related function

ODL List OAM objects

Syntax



collection
Name of the OAM object collection. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

object Name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

```
//S1       EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM OE  COLLECTN=SYSU.OAM.CLLCT001,
$$FILEM     OBJNAME=DTT.ITT.WORK12
$$FILEM EOJ
/*
```

OO (Object to Object)

Purpose
Copy an object

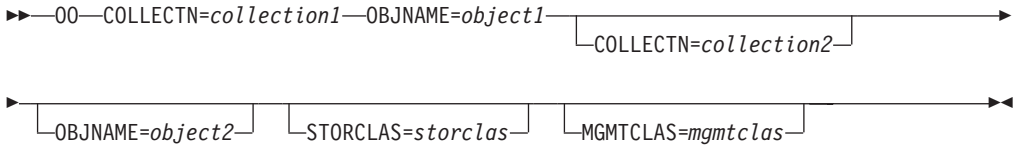
Usage notes
Use this function to copy an OAM object within a collection (using a different object name) or to another collection.

Options
You can specify SMS storage class and management class values to be used for storing the object. These might, however, be overridden by your installation defaults.

Related functions

ODL List OAM objects
OE Erase an OAM object
OS Backup objects from an OAM database to a data set
OV Backup objects from an OAM database to a VSAM data set
OP Print an object in character or hexadecimal dump format
SO Copy a data set to an object database
VO Copy VSAM data to an object database

Syntax



collection1
Name of the OAM object collection that contains the object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

collection2
Name of the OAM object collection to which you are copying the object. In

any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

By default, the collection name is not changed. (In this case, you must change the object name.)

mgmtclas
Name of an SMS management class.

object1
Current name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

object2
New name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

By default, the object name is not changed. (In this case, you must change the collection name.)

storclas
Name of an SMS storage class.

Note: The same keywords are used for input and output. The first occurrences of COLLECTN and OBJNAME refer to the input object; the second occurrences refer to the output object.

```
/* REXX */
/* Copy object within a collection ...       */
 "FILEMGR $OO  COLLECTN='SYSU.OAM.CLLCT000',",
              "OBJNAME='DTT.ITT.WORK12',",
              "OBJNAME='DTT.ITT.COPY12'"
/* Copy object to other collection...        */
 "FILEMGR $OO  COLLECTN='SYSU.OAM.CLLCT000',",
              "OBJNAME='DTT.ITT.COPY12',",
              "COLLECTN='SYSU.OAM.CLLCT001'"
 return
```

OP (Object Print)

Purpose
Print an Object Access Method (OAM) object.

Usage notes
Use this function to print the contents of an object in character format or one of the two dump formats.

Options
You can select the range of bytes or records to print.

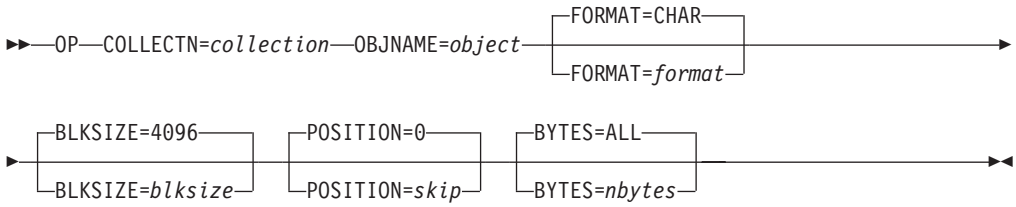
You can use the various SET processing options to control the print output:

- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Related functions

ODL List OAM objects
OE Erase an OAM object

Syntax



blksize
Logical record size to be used to deblock the OAM object into logical records. The maximum is 32 768; the default is 4 096.

collection
Name of the OAM object collection. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

format
The format of the output:
CHAR
Character format (the default)
HEX Hexadecimal format.

nbytes
Maximum number of bytes to be printed or ALL. If this number is exceeded, printing stops. This parameter provides protection against printing an OAM object that is much larger than expected. The maximum number is 99 999 999. If you specify ALL or you omit the parameter, the entire object is printed.

object Name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

skip Number of bytes to be skipped from the beginning of the object. The maximum is 99 999 999; the default is 0.

```
//JOB (acct),'name',Print OAM Object
//S1       EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM OP  COLLECTN='SYSU.OAM.CLLCT000',
$$FILEM     OBJNAME='DTT.ITT.TBL21',
$$FILEM     BYTES=1000
$$FILEM EOJ
/*
```

OS (Object to Sequential Data)

Purpose

Copy one or more OAM objects from a collection, or a primary or backup optical volume, to a sequential data set.

Usage notes

To copy only the data within an object, specify the collection and the object name, and that you do not want a header record.

To create a backup of a single object, specify the collection and the object name, and that you want the header record. File Manager saves the object together with its directory information. Use the SO (Sequential Data to Object) function to restore the object.

To create a backup of multiple objects from a collection, a primary or backup optical volume, specify a generic object name. Specify the collection name, the volser of an optical volume, or both. File Manager saves the directory information of all objects copied. Use the SO (Sequential Data to Object) function to restore all, or selected, objects.

For more information about using this function, see "Backing up OAM objects" on page 386.

Options

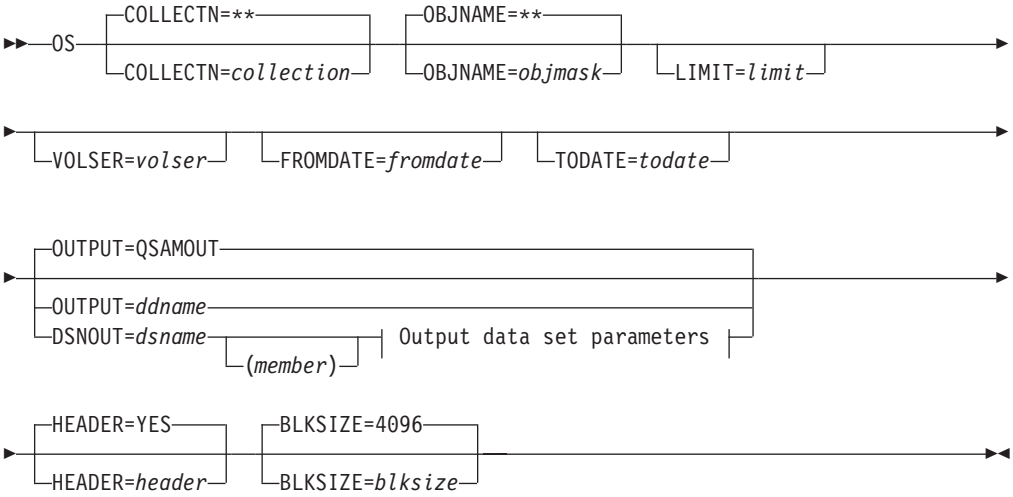
Limit processing to objects residing in DASD storage, on optical storage media, or on a specific optical volume. You can also specify a range for the creation date.

Specify the block size to be used for the output records. The default block size is 4096 bytes, the maximum is 32760 bytes.

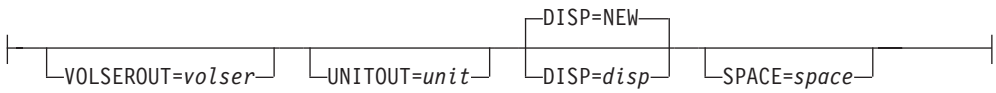
Related functions

OO Copy an object to the same or another collection
OV Backup objects from an OAM database to a VSAM data set
SO Copy a data set to an object database

Syntax



Output data set parameters:



blksize
Block size for the output data set. The maximum is 32 760; the default is 4 096.

collection
A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.

ddname
Refers to a DD or TSO ALLOC statement. The default is QSAMOUT.

disp Disposition of the output data set. Specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.

Note: SMS might modify the allocation of new data sets on your system. For details, contact your SMS Administrator.

dsname
Name of a sequential data set. If specified, any DD statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the syntax diagram.

fromdate
The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.

header
If you are copying only one object, specify NO if you do not want a header record. The default is to put a header record at the beginning of the output data set.

limit To limit the objects that are copied based on where objects are located, specify one of the following:

DASD
Only objects that are stored on DASD

OPTICAL
Only objects that are stored on optical disks

VOLUME
Only objects that are stored on a specified optical disk. You must also specify the volser of the optical disk.

member
For a PDS, the member name.

objmask
An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

space The number of tracks to be allocated for a new data set. Specify *n* or (*n*,*m*), where *n* is the number of primary tracks, and *m* is the number of secondary tracks.

todate The latest creation date, in the form *yyyymmdd*.

unit A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

volser On input: if you specified VOLUME for *limit*, the volume serial number of the optical disk.

On output: the volume serial number for a new or non-cataloged data set.

```
// JOB (acct),'name',Save collection on tape
//S1       EXEC PGM=FILEMGR
//BACKUP   DD DSN='OBJECT.BACKUP',UNIT=CARTR,
//            DISP=(NEW,CATLG)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM OS  COLLECTN=SYSU.OAM.CLLCT000,
$$FILEM     OUTPUT=BACKUP
$$FILEM EOJ
/*
```

OV (Object to VSAM)

Purpose

Copy one or more OAM objects from a collection or a primary or backup optical volume to a VSAM data set.

Usage notes

To copy only the data within an object, specify the collection and the object name, and that you do not want a header record.

To create a backup of a single object, specify the collection and the object name, and that you want the header record. The output data set must be a VSAM ESDS. File Manager saves the object together with its directory information. Use the VO (VSAM to Object) function to restore the object.

To create a backup of multiple objects from a collection, a primary or backup optical volume, specify a generic object name. Specify the collection name, the volser of an optical volume, or both. The output data set must be a VSAM ESDS. File Manager saves the directory information of all objects copied. Use the VO (VSAM to Object) function to restore all or selected objects.

For more information about using this function, see “Backing up OAM objects” on page 386.

Options

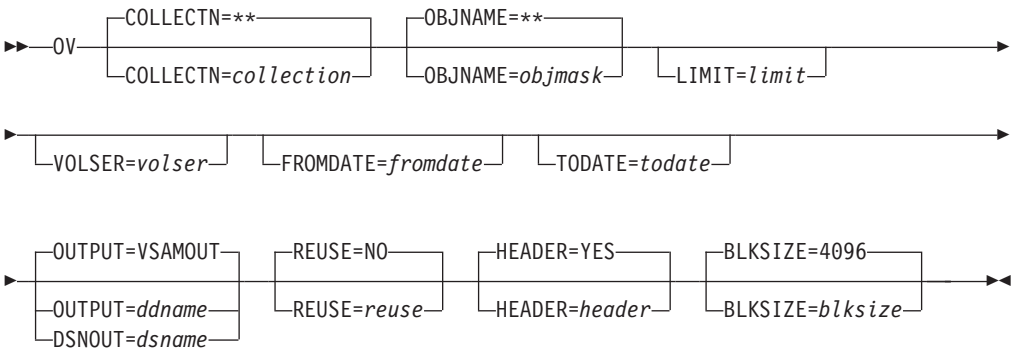
Limit processing to objects residing in DASD storage, on optical storage media, or on a specific optical volume. You can also specify a range for the creation date.

Specify the block size to be used for the output records. The default block size is 4096 bytes, the maximum is limited by the maximum record size defined for the VSAM cluster.

Related functions

OO Copy an object to the same or another collection
OS Backup objects from an OAM database to a data set
VO Copy VSAM data to an object database

Syntax



blksize
Block size for the output data set. The maximum depends on the cluster definition; the default is 4 096.

collection
A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.

ddname
Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.

dsname
Name of a VSAM data set.

fromdate
The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.

header
If you are copying only one object, specify NO if you do not want a header record. The default is to put a header record at the beginning of the output data set.

limit To limit the objects that are copied based on where objects are located, specify one of the following:

DASD
Only objects that are stored on DASD

OPTICAL
Only objects that are stored on optical disks

VOLUME
Only objects that are stored on a specified optical disk. You must also specify the volser of the optical disk.

objmask
An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

reuse Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

todate The latest creation date, in the form *yyyymmdd*.

volser If you specified VOLUME for *limit*, the volume serial number of the optical disk.

```
/* REXX */
/* This EXEC saves all OAM objects in a VSAM */
/* data set which were added to a collection */
/* at the current day:                       */

/* Get date in format YYYYMMDD               */
fromdate = DATE('STANDARD')
say 'Saving objects from' fromdate

/* create backup copy of all new objects     */
"FILEMGR $OV  COLLECTN='SYSU.OAM.CLLCT000',",
             "FROMDATE="||fromdate,
             "DSNOUT=REUSABLE.ESDS,REUSE=YES"

 return
```

PBK (Print Copybook)

Purpose

Prints a report showing the field definitions in:

- a copybook
- a template that was based on a copybook (not DB2 templates)
- a dynamic template
- an IMS view
- an IMS criteria set

Usage notes

When referencing a copybook, the copybook must be stored in a partitioned data set member, a CA-Panvalet library member, or a library accessed using the Library Management System Exit. All other source types can be stored in a sequential data set, partitioned data set member, CA-Panvalet library member, or a library accessed using the Library Management System Exit.

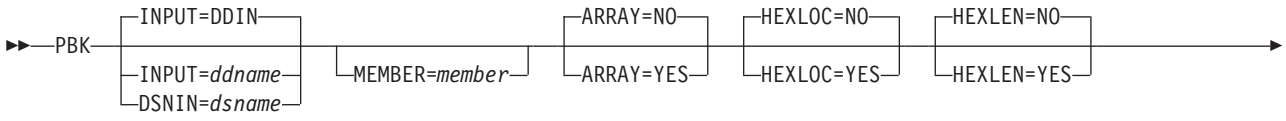
Options

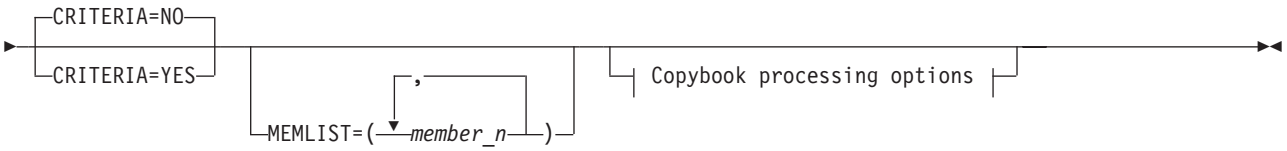
None.

Related functions

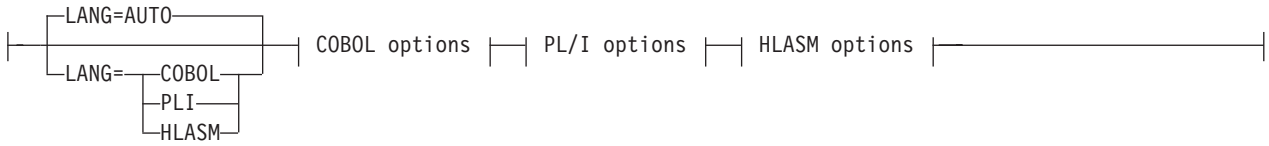
None.

Syntax

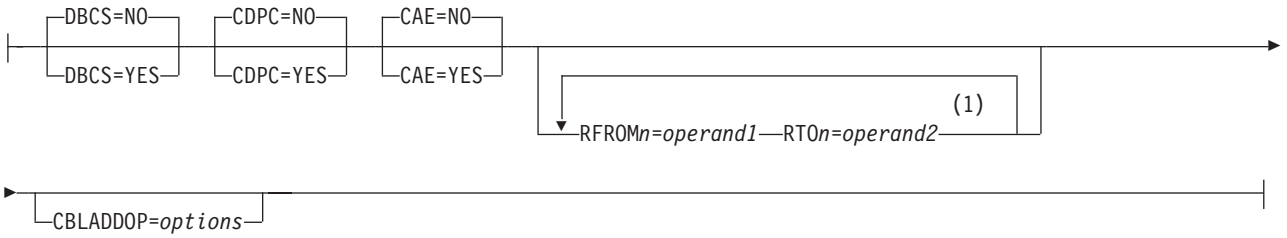




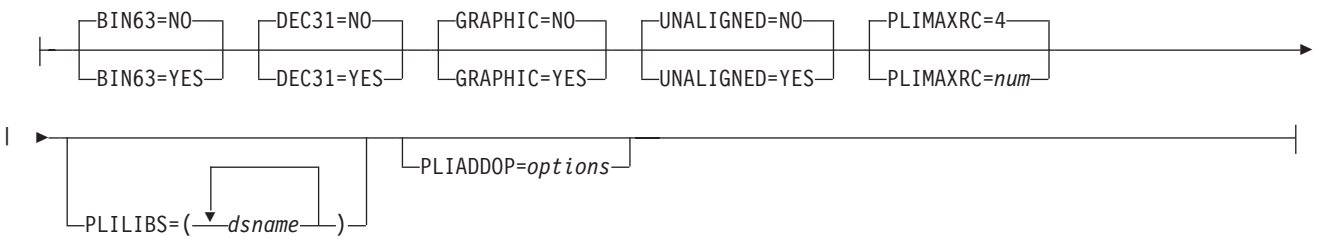
Copybook processing options:



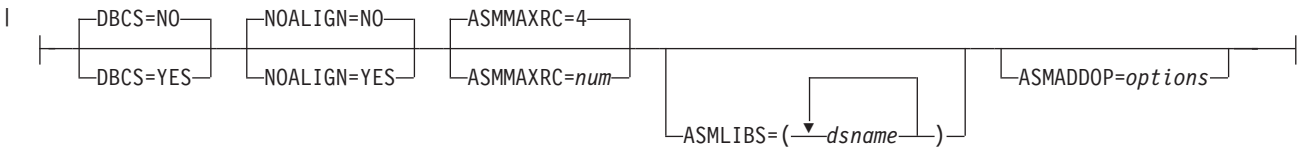
COBOL options:



PL/I options:



HLASM options:



Notes:

1 *n*= 1 to 5.

INPUT=*ddname*
Defines a reference to a DD or TSO ALLOC statement for the input data sets. The default is DDIN.

If you specify a concatenated DD, then you must provide the member name, *member*.

DSNIN=*dsname*
Specifies the name of the data set that contains the copybooks, templates, IMS views, or IMS criteria sets. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty.

MEMBER=*member*
Where *dsname* is a partitioned data set, CA-Panvalet library, or a library accessed using the Library Management System Exit, specifies the member name. You can specify a pattern to print multiple members. If the input data set is partitioned and you omit the MEMBER parameter, then every member is printed.

ARRAY=YES
Print all occurrences of array elements.

ARRAY=NO
Do not print all occurrences of array elements.

HEXLOC=YES
Print start and end values as hexadecimal offsets.

HEXLOC=NO
Print start and end values as decimal offsets.

HEXLEN=YES
Print length values as hexadecimal.

HEXLEN=NO
Print length values as decimal.

CRITERIA=YES
Both template layout and template criteria information are reported.

CRITERIA=NO
Only template layout information (and not template criteria information) is reported.

MEMLIST
Allows you to specify a list of member names.

member_n
The name of the member to be processed. Generic name masks are allowed.

INPUT=DDIN

Note: If you specify a concatenated DD, then you must provide the member name, *member*.

Copybook processing
If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG
Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL, PL/I, or HLASM.

AUTO
Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If

the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL
Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

HLASM
Invoke the HLASM compiler to create a template from the copybook.

COBOL options
The following options are used to compile a COBOL copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO
Do not use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CDPC = YES
Use the COBOL SPECIAL-NAMES paragraph "Decimal-point is comma".

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

MIXED = NO
Field names stored in the template in uppercase.

MIXED = YES
Field names stored in the template in the original case as coded in the COBOL copybook.

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying "From" and "To" strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

CBLMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the CBLMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

CBLLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

CBLADDOP
Additional COBOL compiler options which are included in a CBL statement when compiling COBOL copybooks.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES
Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO
Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES
Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO
Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

PLIMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the PLIMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

PLILIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

PLIADDOP
Additional PL/I compiler options which are included in a *PROCESS statement when compiling PL/I copybooks.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

HLASM options
The following options are used to compile a HLASM copybook into a template:

DBCS=YES
Use the DBCS compiler option.

DBCS=NO
Use the NODBCS compiler option.

NOALIGN=YES
Use the NOALIGN compiler option.

NOALIGN=NO
Use the ALIGN compiler option.

ASMMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

Note: The COMPMAXRC parameter is still supported but it is recommended that the ASMMAXRC parameter is used instead. If you do specify the COMPMAXRC parameter, it takes precedence over the language MAXRC.

ASMLIBS
Allows you to specify a list of up to ten data set names to be specified in the SYSLIB concatenation list. These data sets are searched in the order specified for COPY and INCLUDE members for the compilation.

dsname
The name of the data set name to be processed. Generic name masks are not allowed.

ASMADDOP
Additional HLASM compiler options which are included in a *PROCESS statement when compiling COPY and MACRO members.

For details on the effect of these compiler options, see the *HLASM V1R5 Programmer's Guide*.

```
$$FILEM PBK DSNIN=FMNUSER.TEMPLATE,
$$FILEM MEMBER=DITTST1
```

REW (Tape Rewind)

Purpose
Rewind a tape to the load point.

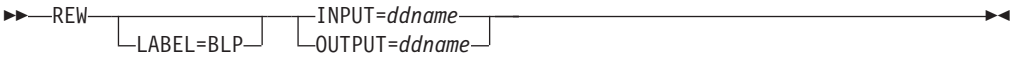
Usage notes
After completion of the function, the tape is positioned at load point (beginning of tape).

Options
None.

Related functions

BSF Move the tape backward one or more files
BSR Move the tape backward one or more records
ERT Erase a tape from the current position to the end of the tape
FSF Move the tape forward one or more files
FSR Move the tape forward one or more records
RUN Rewind a tape and unload it
WTM Write one or more tape marks

Syntax



ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

RUN (Tape Rewind-Unload)

Purpose
Rewind a tape and unload it.

Usage notes
Use this function to rewind a tape and unload it. The tape is closed; the allocation is freed if the tape was dynamically allocated during the current File Manager session.

Options

None.

Related functions

BSF Move the tape backward one or more files
BSR Move the tape backward one or more records
FSF Move the tape forward one or more files
FSR Move the tape forward one or more records
REW Rewind a tape to its load point

Syntax

ddname

Refers to a DD or TSO ALLOC statement.

SCS (Catalog Services)

Purpose

Print a list of catalog entries.

Usage notes

Use this function to print catalog entries.

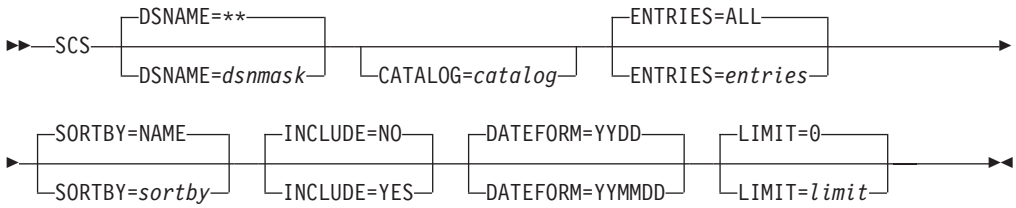
Options

Use a fully-qualified or generic file ID to limit the output. You can also specify the type of objects you want to see. The list can be sorted by entry name, creation date, expiration date, or amount of free space.

Related function

DVT Print VTOC entries

Syntax



CATALOG

Name of a catalog. By default, the system catalog search order is used.

dsnmask

A generic data set name. Within the data set name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers. The default is **.

To print a single entry, specify a fully-qualified entry name.

ENTRIES

The entries you want listed:
ALL All catalog entries (the default)

ALLVSAM
All VSAM catalog entries
ALIAS
Alias entries
AIX Alternate index entries
CLUSTER
Cluster entries
DATA VSAM data entries
GDG Generation data group entries
INDEX
VSAM index entries
NONVSAM
NonVSAM entries
OAM OAM collection entries
PATH VSAM path entries
PGSPC
Pagespace entries
UCAT User catalog entries.

SORTBY
One of the following:
ALLOC
Sorts by allocated space
DATE Sorts by creation date
DSORG
Sorts by organisation
FREESP
Sorts by free space
LRECL
Sorts by logical record length
NAME
Sorts by data set name (the default)
RECFM
Sorts by record format

INCLUDE
One of the following:
NO Restricts the data set list to data sets with only the qualifiers entered in the **DSNAME** field. Data sets with additional qualifiers are not be included.
YES Generates the data set list with all data sets matching the qualifiers in the **DSNAME** field, including data sets with additional qualifiers.

DATEFORM
One of the following:
YYDDD
All dates on the Data Set List panel and listing shown as YYYY.DDD.
YYMMDD
All dates on the Data Set List panel and listing shown as YY/MM/DD.

limit The maximum number of data sets selected for VTOC or catalog processing.

```
//SCS JOB (acct),'name'  Catalog Services
//*
//FMBAT    PROC
```

```
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//DELETE   EXEC PGM=IEFBR14
//FILE     DD DSN=FMNUSER.SCS.LISTING,
//            DISP=(MOD,DELETE),
//            SPACE=(TRK,(30,10)),UNIT=SYSDA,
//            RECFM=FBA,LRECL=133
//*
//SCS00    EXEC FMBAT ,REGION=6M
//SYSPRINT DD DSN=FMNUSER.SCS.LISTING,
//            DISP=(,CATLG),
//            SPACE=(TRK,(30,10)),UNIT=SYSDA,
//            RECFM=FBA,LRECL=133
//SYSIN DD *
$$FILEM SCS
$$FILEM EOJ
/*
```

SET (Set Processing Options)

Purpose

Set various File Manager processing control options.

Usage notes

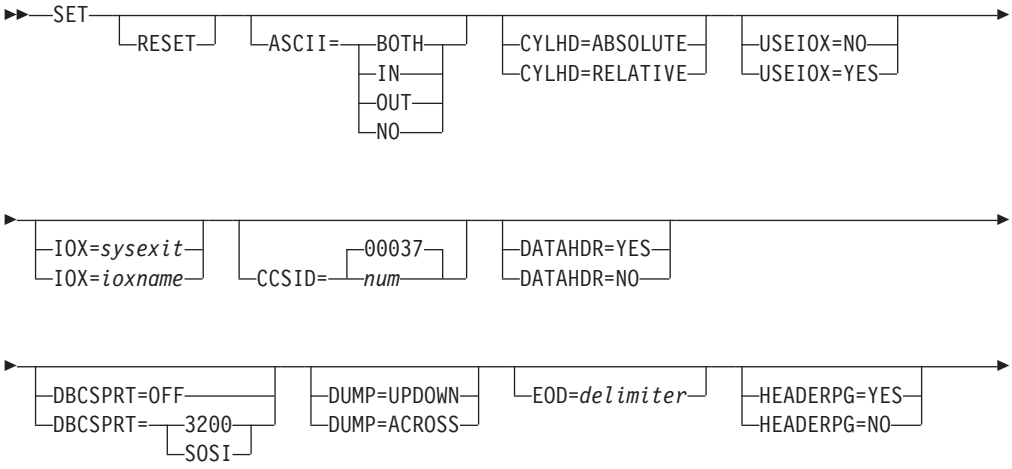
Use this function to list the processing options that are currently in effect. You can also use the SET function to change one or more options, or reset options to the installation default. If you use the SET function within a batch job, your changes affect only the current batch job. In any other case, your changes are saved in the user profile and are available for further File Manager sessions.

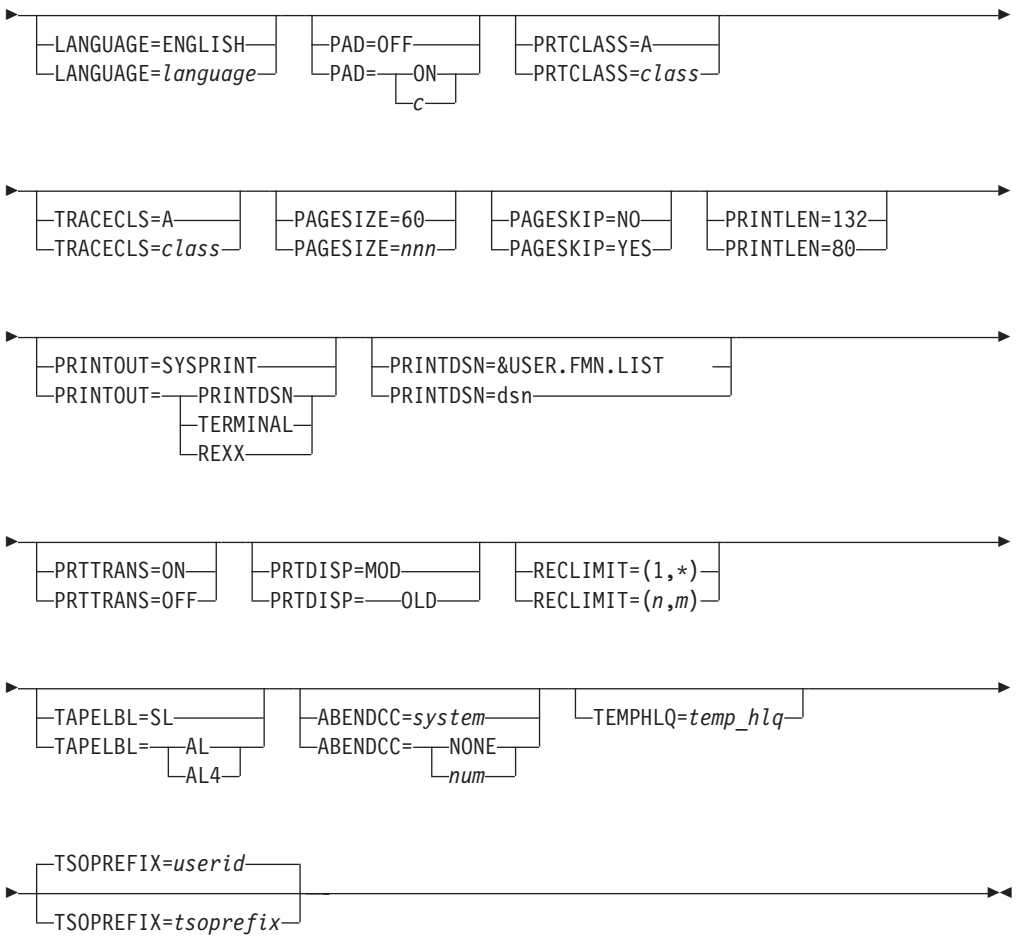
To change the installation defaults that apply to *all* users and batch jobs, see the *File Manager Customization Guide*.

Options

None.

Syntax





Enter SET with no parameters to display the current settings.

RESET
Returns all options to the installation default values (described in the *File Manager Customization Guide*).

ASCII Specifies requirements for translation of tape data between ASCII and EBCDIC. The default setting for this parameter is determined at installation, by the value of the ASCII parameter in the FMN0POPT macro. Typically, this is set to BOTH.

BOTH Translates tape input from ASCII format to EBCDIC format, and translates tape output from EBCDIC format to ASCII format.

IN Translates tape input from ASCII format to EBCDIC format.

NO Does not translate data.

OUT Translates tape output from EBCDIC format to ASCII format.

This processing option applies to all tape input and output functions except: Initialize Tape, Tape Browse, Tape Label Display, and Tape Update. For Tape Label Display, ASCII translation is automatically performed when needed. For Initialize Tape and Tape Browse, you can specify ASCII input or output when invoking the function. For the other (non-tape) functions, ASCII translation is not supported by File Manager.

CYLHD

Defines the way you specify the location of a disk data set in disk functions.

ABSOLUTE

File Manager interprets cylinder-head and track-number values as actual physical addresses.

RELATIVE

File Manager interprets cylinder-head and track-number values as relative to the start of the data set.

USEIOX

Specifies whether to invoke a user I/O exit.

Note:

1. If your site customization for File Manager does not allow user I/O exits, you cannot use this option to override that setting.
2. The user I/O exit values set with the SET function have no effect unless a subsequent function requests I/O exit processing using its own options; for example, USEIOXIN for the DSC (Data Set Copy) input user I/O exit.

Also, see "Setting your System Processing Options (option 0.2)" on page 43.

NO Do not invoke a user I/O exit.

YES Invoke a user I/O exit. This option is only effective if your site customization for File Manager allows user I/O exits on a site-wide basis.

IOX Specifies the name of the user I/O exit. This option only takes effect if your site customization for File Manager allows user I/O exits on a site-wide basis, *and* one of these is true:

- A prior SET function set USEIOX to YES.
- The current SET function is setting USEIOX to YES.

There are no restrictions on the programming language that you can use to write an exit, however, the exit must be provided to File Manager in the STEPLIB/ISPLLIB concatenation or their extensions (LINKLIST, LPA, and so on).

sysexit Default. If you specify USEIOX=YES and do not supply a user I/O exit name, File Manager uses the name of the exit provided by either a prior SET function in the current job step, or the installation customization options. If you specify USEIOX=YES and no installation default has been provided, you must specify IOX=*ioxname*.

Note: If you have selected batch processing in an online panel, the generated JCL statements use the default name provided in your Set System Processing Options panel.

ioxname

The name of a PDS(E) member of a data set that has been provided to File Manager in the STEPLIB concatenation.

CCSID

Specifies the CCSID to be used as the default CCSID when File Manager is run in batch. The CCSID specified should be a valid CCSID and a CCSID which is recognised by the Z/OS system which runs the batch job.

DATAHDR
When data is printed in CHAR print format, specifies whether record number and length are included in character-format print output.

The DATAHDR setting affects only DSP and FCH processing.

NO Record header information is not included in character-format output. Specify DATAHDR=NO to left-justify the data in the print output, without header information.

YES Record header information is included in character-format output.

DBCSPRT
This option is no longer used. The syntax is kept for compatibility only.

DUMP
Specifies the format of hexadecimal print output (for example, when you use Tape Print with **Print format** set to HEX). For an example of each format, see "Selecting a display format" on page 66.

ACROSS
The hexadecimal digits are displayed as 2 groups of 4 full words resulting in 32 hexadecimal digits followed by the EBCDIC character display to the right of the hexadecimal display.

UPDOWN
The two digits making up the hexadecimal representation of each EBCDIC character are displayed vertically directly under that character.

EOD End-of-data delimiter for tape input.

delimiter
A 1- to 8-character delimiter. For character data, enclose the string in quotation marks if it contains blanks, commas, or lowercase letters. For hexadecimal data, enter an X followed by the string enclosed in quotation marks (for example, X'04').

For some tape functions, you specify the number of files to process in an NFILES keyword. If you specify NFILES=EOD, then files are processed until the first record that begins with the EOD delimiter. However, there is no default EOD delimiter for tape input. So, if you want to specify NFILES=EOD, then you must first use the SET function to specify an EOD delimiter; otherwise, File Manager issues an error.

HEADERPG
Specifies whether a header page (a title page with File Manager on it) is included in print output. If the PRINTOUT option is set to TERMINAL, then the header page is not included in print output, even if HEADERPG is set to YES.

NO A header page is not included in print output.

Note: For the HEADERPG=NO setting to be effective for the duration of the entire batch processing, you must specify it as the first File Manager control statement. The HEADERPG=NO setting also suppresses the output of the message "FMN0062I current SET function values" as long as it is active.

YES A header page is included in print output.

LANGUAGE
Specifies the language of File Manager message text and panel text. The possible values depend on which translated message text you installed.

For instructions on installing translated messages and panels, and the keyword values that you can specify for the LANGUAGE processing option, see the *File Manager Customization Guide*.

PAD Specifies whether records are padded or truncated while being copied.

OFF Records are not padded with blanks or a specified character, except where required by the output record format. If the output records are fixed format, they are padded with nulls ('00'x), as required to match the output record length. Variable and undefined format outputs are not padded.

Output records are always truncated if they exceed the maximum record length (for fixed or variable) or maximum block length (for undefined formats). However, trailing blanks that fit within the maximum length are not stripped from the record.

ON Records are modified, depending on the output record format. For fixed-format output, input records that do not fit are truncated or padded with blanks. For variable or undefined output, trailing blanks are stripped from the input record. (For reasonable results, do not try to deblock and pad in one run.)

c The value used for padding records. This can be a single character or a hexadecimal value. For example, you can enter an `X` followed by two hexadecimal characters enclosed in quotation marks (X'00'). In addition, for variable or undefined output, a trailing series of the specified character is stripped from the input record. Essentially, this option has the same effect as PAD=ON except that the specified character *c* is used for padding or truncation instead of blanks.

PRTCLASS
Specifies the class of the JES spool queue to be used when the PRINT command is issued while browsing the temporary print data set.

TRACECLS
Specifies the class of the JES spool queue to be used when the PRINT command is issued when browsing the temporary trace data set.

PAGESIZE
Specifies the number of lines printed on each page of print output. You can specify a value from 1 to 999. The default is 60.

PAGESKIP
Specifies whether print output from each function begins on a new page.
NO Print output from each function does not begin on a new page.
YES Print output from each function begins on a new page.

PRINTLEN
Specifies the line length of print output.
80 The line length of print output is 80 characters, suitable for a terminal.
132 The line length of print output is 132 characters, suitable for a printer.

Note: PRINTLEN is not applicable when:
- The data is printed in TABL format, or
- You specify the wide format for the compare output listing

In either of the above cases, when data is printed the line length is determined from the number and size of fields printed. If this length is greater than the record length specified for the data set, the print line is truncated. The maximum print line length is 32760.

PRINTDSN
If PRINTOUT=PRINTDSN (see below) is specified, then File Manager sends print output to this temporary data set. The default is *userid*.FMN.LIST, unless changed in your File Manager installation.

To browse this data set, enter PB on any File Manager Command line. Then, to forward the contents of the data set to PRTCLASS class, enter PRINT. For details, see "Printing from File Manager" on page 299.

PRINTOUT
Specifies where print output is sent, except for batch execution.

SYSPRINT
Send print output to the current SYSPRINT allocation.

TERMINAL
Send print output to the TSO terminal.

PRINTDSN
Output is accumulated in the print data set specified in the Data set name field (PRINTDSN). This data set can browsed using the PB command or sent for printing to the JES spool queue with class specified in the Output class field using the PRINT command while browsing.

REXX Send output to a REXX stem variable. Each line corresponds to a variable named FILEM.*nnn*. FILEM.0 contains the line count.

When you use the PRINT primary command in Print Browse (option 3.9), or if you specify batch execution in a File Manager panel, then print output is always sent to a JES class.

PRTTRANS
Specifies how File Manager translates non-printable characters.

OFF No translation performed. Use PRTTRANS=OFF to support special print chains. This might make printing faster.

ON Non-printable characters translated to blanks using a translate table.

For instructions on altering the print translate table, see the *File Manager Customization Guide*.

PRTDISP
Specifies the disposition of a print data set.

OLD The print data set is cleared before each print operation, and print output is written from the beginning of the data set.

MOD Default setting. Print output is appended to the existing print data set.

RECLIMIT
When data is printed in CHAR or LHEX print format, limits print output for each record (or OAM object).

(*n*,*m*) Print output is limited to the data from columns (bytes) *n* through *m*. An asterisk (*) specified for *m* indicates the end of the record.

(1,*) The entire record (or block) is printed.

TAPELBL

Specifies the type of labels that are created:

AL ANSI Version 3 labels are created. ("Version 3" refers to ANSI X3.27–1978, ISO 1001–1979, and FIPS 79 standards.)

AL4 ANSI Version 4 labels are created. ("Version 4" refers to ANSI X3.27–1987 level 4 and ISO 1001–1986(E) standards.)

SL EBCDIC labels are created.

For detailed information about ANSI tape labels, see *z/OS DFSMS: Using Magnetic Tapes*.

This parameter only affects Initialize Tape (option 4.12).

ABENDCC

Specifies how File Manager processing resulting in non-zero return code (including customized return code) is completed in batch.

system Default. This option can be set as part of the customization options for your site. If you do not specify the ABENDCC parameter in the SET function, the installation setting is used. The installation setting defaults to NONE, but if it is set as an integer, that integer becomes the default setting for File Manager batch processing.

NONE

The job step is finished with a non-zero return code.

num Must be specified as an integer, from 1 to 32767 (0 is treated as NONE).

Use this parameter to transform an original or customized non-zero return code to an abend. If the return code is greater than or equal to *num*, File Manager abends with U999, Reason Code=888 (hex: 378). Abnormal termination of a File Manager step prevents the execution of successors (steps and/or jobs). You can specify the *num* value appropriately to suit the customized return codes issued by File Manager functions.

Note: In addition, specification of any non-zero integer prevents File Manager from intercepting abnormal terminations (system abends).

TEMPHLQ

TEMPHLQ specifies an installation-specific high-level qualifier for the temporary data sets created during a File Manager session. You can specify any valid qualifier, including multilevel if required, up to a total length of 24 characters. If you specify &USER. for any part of the qualifier, this is replaced with the user's userid. A multilevel qualifier with symbols follows the same rules as in the specification of a data set name with symbolic parameters in JCL procedures. Thus, a single period used after a symbol combines the symbol with the text following it (See Example 1 below); a period separating parts of a multilevel qualifier must be doubled (see Example 2 below).

Example 1

If the userid is XXXX, then specifying &USER.1.TEMP results in a high-level-qualifier of XXX1.TEMP

Example 2

If the userid is ZZZZ then specifying FMN.&USER..ABCD results in a high-level-qualifier of FMN.ZZZZ.ABCD

Note:

1. The TEMPHLQ parameter can be up to 24 characters long. However, temporary data sets can have different lengths, and the system part of the data set name may be longer than 20 characters. In this case, minor levels of TEMPHLQ are ignored, to the extent that the final data set name is less than or equal to 44 characters.
2. Concatenation of &USER.string could result in a part of the high-level qualifier being longer than 8 characters. In this case, the string is shortened to the extent that the final part of the qualifier is 8 characters long.

TSOPREFIX

This parameter allows you to specify the user's TSO prefix. The parameter is provided because functions run in batch mode cannot obtain the user's TSO prefix but sometimes they require it. Batch functions may require it when the &PREFIX symbol is specified in the TEMPHLQ or AUDITHLQ parameters in the File Manager Installation Options modules or the TEMPHLQ parameter of a SET function. The &PREFIX symbol represents the user's TSO prefix, so when File Manager functions encounter this symbol in any of the above parameters, they are supposed to substitute the user's TSO prefix. Batch functions can only do this, if this parameter is specified.

SO (Sequential Data to Object)

Purpose

Restore one or more OAM objects that were backed up in a sequential data set, or create a new object from a sequential data set.

Usage notes

Use this function to copy data from a QSAM data set to an object database. If the input is a backup data set created with the OS (Object to Sequential Data) function, the object header records are used for restoring.

If loading from an object backup data set the objects are restored with the original name into the original collection. To filter loading specify a collection name, a generic object name, the location (DASD or OPTICAL, or a specific optical volume), and a creation date range. The filter applies to the object header information saved with the objects.

If the input does not contain backup copies of multiple objects, one object is created. You can specify a collection name and an object name. If the input data begins with an object header record, the default values are taken from it. The object is made up of the concatenated input records, excluding any object header.

Options

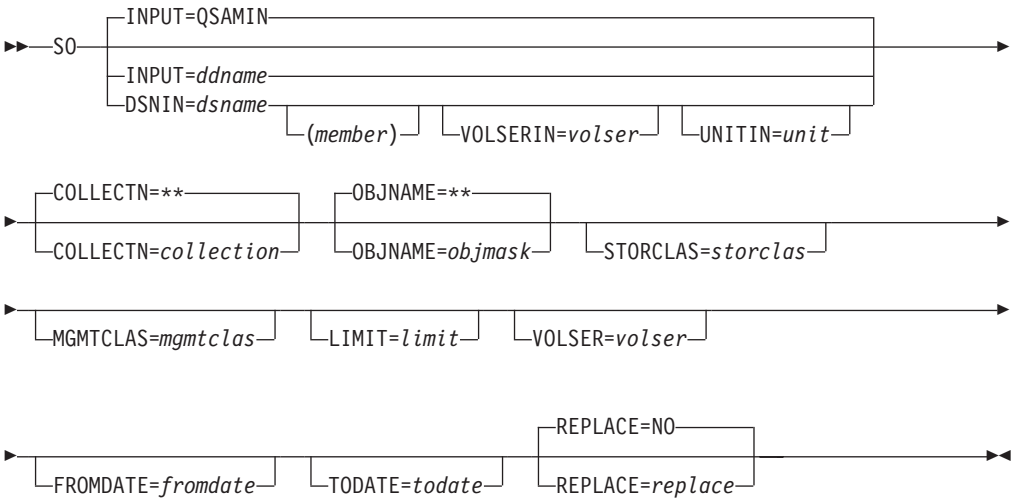
You can specify SMS storage class and management class values to be used for storing the object; these might, however, be overridden by your installation defaults. You can also specify whether an existing object with a matching name is to be replaced in the output collection.

For more information about using this function, see “Restoring OAM objects” on page 388.

Related functions

OO Copy an object to the same or another collection
OS Backup objects from an OAM database to a data set
VO Copy VSAM data to an object database

Syntax



collection

If the data set contains one object with a header, you can optionally specify a collection name to override the collection name in the header.

If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify the collection where the object is stored.

If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify a collection name to restore only objects from the specified collection.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

ddname

Refers to a DD or TSO ALLOC statement. The default is QSAMIN.

dsname

Name of a sequential data set. If specified, any DD statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the syntax diagram.

fromdate

The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.

limit To limit the objects that are copied based on where the objects were located (before they were backed up to a data set), specify one of the following:

DASD

Only objects that were stored on DASD

OPTICAL

Only objects that were stored on optical disks

VOLUME
Only objects that were stored on a specified optical disk. You must also specify the volser of the optical disk.

member
For a PDS, the member name.

mgmtclas
Name of an SMS management class.

objmask
If the data set contains one object with a header, you can optionally specify an object name to override the object name in the header.

If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify a name for the object.

If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify an object name or generic object name to limit the objects that are restored based on object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) with any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

replace
Specify YES to replace any existing objects with the same name in the collection. By default, the object is not copied.

storclas
Name of an SMS storage class.

todate The latest creation date, in the form *yyyymmdd*.

unit A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

volser On input: the volume serial number for a non-cataloged data set.

On output: if you specified VOLUME for *limit*, the volume serial number of the optical disk.

```
//SO  JOB (acct),'name' Restore Objects
//FILEMGR  EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//OBJBKUP  DD DSN='FMN.OBJECT.BKUP',
//            VOL=SER=FMO010,UNIT=CARTR,
//            DISP=OLD
//SYSIN    DD *
$$FILEM SO INPUT=OBJBKUP,OBJNAME=**,
$$FILEM    COLLECTN=SYSU.OAM.CLLCT000,
$$FILEM    LIMIT=VOLUME,VOLSER=OVOL03
$$FILEM EOJ
/*
```

ST (Sequential Data to Tape)

Purpose
Copy records from a sequential data set to a tape file.

Usage notes

Use this function to copy records from a sequential data set to an unlabeled tape.

Unless you have strong reasons for not doing so, use bypass label processing (BLP) with this function.

Specify a DDNAME for the tape.

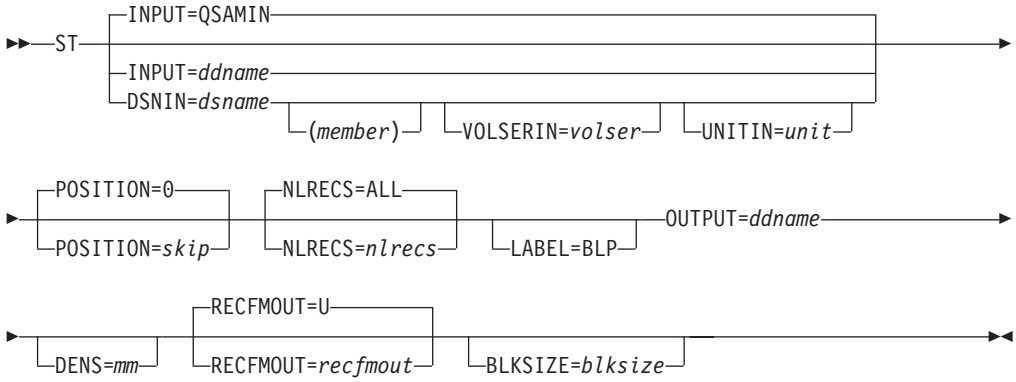
Options

You can specify the number of records to skip from the beginning of the input data set. You can also specify the number of records to copy, and the output record format and block size.

Related functions

DSC Copy a data set to another one
TLT Copy multivolume, multiple-file labeled tapes
TS Copy tape data to a data set

Syntax



blksize

If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape is processed by other utilities or standard access methods, you must also consider the operating system limits.

ddname

Refers to a DD or TSO ALLOC statement. The default is QSAMIN for input. To create the output tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

dsname

Defines the name of the data set. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. To further describe the data set, use the appropriate keywords as shown in the syntax diagram.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *File Manager Customization Guide*.

member
For a PDS, the member name.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

recfmout
Record format for the output. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

skip Number of logical records to be skipped from the beginning of the data set. The default is 0.

unit A device number. A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

volser Volume serial number for a non-cataloged data set.

```
//ST  JOB (acct),'name'
//* Copy Sequential file to SL Tape
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//DELETE   EXEC PGM=IEFBR14
//BSA1     DD DISP=(MOD,DELETE),UNIT=SYSALLDA,
//            SPACE=(TRK,(1,1),RLSE),
//            DSN=FMNUSER.BS43FB
//*
//STP1001  EXEC FMBAT
//QSAMOUT  DD DISP=(,CATLG),UNIT=SYSALLDA,
//            SPACE=(CYL,(1,1),RLSE),
//            RECFM=FBS,DSN=FMNUSER.BS43FB
//OUTTAPE  DD DSN=TDALAIS.TAPEOUT,UNIT=CART,
//            DISP=(,KEEP),LABEL=(2,SL),
//            VOL=(,RETAIN,,,SER=FMO001)
//SYSIN    DD *
$$FILEM VER
$$FILEM DSG OUTPUT=QSAMOUT,NLRECS=5677,
$$FILEM     RECSIZE=800,KEYLOC=1,KEYLEN=5,
$$FILEM     INCR=5,RECFMOUT=FB,FILLCHAR=AN
$$FILEM ST  INPUT=QSAMOUT,OUTPUT=OUTTAPE,
$$FILEM     RECFMOUT=VB,BLKSIZE=6000
$$FILEM EOJ
/*
```

TLB (Tape Label Display)

Purpose

Print tape labels and a tape label summary.

Usage notes

Use this function to print header and trailer label records for all the data sets on a tape.

If a tape is labeled (or if you want to find out if it is labeled), use the TLB function to print the tape labels. File Manager scans the tape until end-of-volume and prints the header and trailer label records for all files on the tape. A summary of the label data is printed at the end of the listing.

Specify the unit address of the tape and optionally a tape mode.

If you do not know the volser and first data set name of the tape, specify BLP during allocation. If your installation does not allow BLP usage, but File Manager is running APF-authorized, File Manager forces BLP for this function; for security reasons, the tape is freed (unallocated) when TLB ends. You must code a deferred mount in the DD statement.

For further information on the use of this function, see "Tape Label Display (option 4.8)" on page 361.

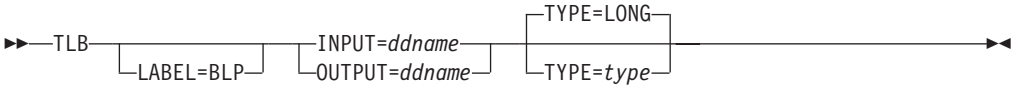
Options

You can choose a short listing, which prints only the VOL1 label and the first header label set.

Related functions

DSP Print records from a data set
TMP Print a summary of the tape contents
TP Print a specified number of blocks from one or more files on a tape

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

type Specify one of the following:

LONG

Print all header and trailer label records for all data sets on the tape (the default).

SHORT

Print only the VOL1 label and the header labels for the first data set.

```
//TLB JOB (acct),'name'  Tape Label Display
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
```

```
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=561,VOL=SER=FMO001,
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM TLB INPUT=TAPE
$$FILEM EOJ
/*
```

TLT (Tape to Labeled Tape)

Purpose

Copy standard labeled tapes with label processing.

Usage notes

Use this function to copy standard labeled tapes (without requiring you to specify every file ID). Labels read from the input tape are interpreted and copied to the output tape. Input from, and output to, multivolume and multiple file tapes is supported.

You can copy all of the files on the tape (the default), or specify the number of files that you want to copy. Copying starts at the current tape position. This lets you select files from the input volume, and append output volumes.

For an input tape containing both labeled and unlabeled files, specify a MIXED tape format. By default, File Manager ends the function and indicates an error when the input tape does not adhere to the IBM label standards.

File Manager might force BLP when switching to the next volume. If BLP is forced, for security reasons the tape is freed (unallocated) when TLT ends.

In some error conditions (for example, if unexpired output volumes are used), prompt messages are sent to the operator console.

Specify a DDNAME for the tape.

For further information on the use of this function, see “Tape to Labeled Tape (option 4.2.3)” on page 350.

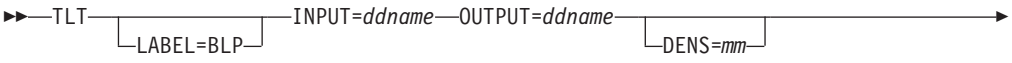
Options

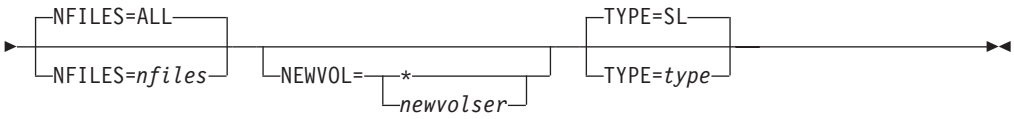
You can specify a list of new volume serial numbers to be used to rename (initialize) the output volumes dynamically. Specify an asterisk (*) instead of new serial numbers to copy the serial numbers of the input volumes. By default, the serial numbers of the output volumes are not changed.

Related functions

TT Copy files from one tape to another without label processing
TTC Compare two tapes byte by byte
TTR Copy a file from one tape to another with reblocking

Syntax





ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

newvolser
Serial number (VOL1) for the initialization of the output volumes. Specify up to five serial numbers in the form (*volser1*, *volser2*,...) or specify an asterisk (*) to copy the volume serial numbers of the input volumes. If omitted, the serial numbers of the output volumes are not changed.

nfiles Number of tape files to be copied or ALL. The maximum is 99 999; the default is ALL.

type Specify one of the following:
MIXED
When copying a tape containing both labeled and unlabeled files.
SL When copying a standard labeled tape (the default). File Manager indicates an error if the input tape does not have IBM standard labels.

```
//TLT JOB (acct),'name'  Tape to Labeled Tape
//FILEMGR   EXEC PGM=FILEMGR
//SYSPRINT  DD SYSOUT=*
//TAPE1     DD UNIT=(381,,DEFER),LABEL=(,BLP),
//             VOL=SER=(FMO1,FMO2),DISP=OLD
//TAPE2     DD UNIT=(382,,DEFER),LABEL=(,BLP),
//             VOL=SER=(FMO3,FMO4),DISP=OLD
//SYSIN     DD *
$$FILEM TLT INPUT=TAPE1,OUTPUT=TAPE2,NFILES=ALL
$$FILEM EOJ
/*
```

TMP (Tape Map)

Purpose
Summarize tape contents.

Usage notes
Use this function to determine the contents of a tape. For a specified number of files on the tape, this function prints the contents of the first block or blocks, the number of records, the minimum and maximum block sizes, the size of the file, and the amount of tape used.

You can summarize the contents of the entire tape, or you can limit the summary to a specific number of tape files and, within each tape file, to a specific number of blocks.

Printing starts with the record following the current tape position and includes any leading tape mark. Processing stops after the specified number of files.

Specify a DDNAME for the tape.

For further information about the use of this function, see "Tape Label Display (option 4.8)" on page 361.

Options

Specify the number of files to be mapped, the number of blocks to be printed from each file, and whether to print in character format or one of the two dump formats.

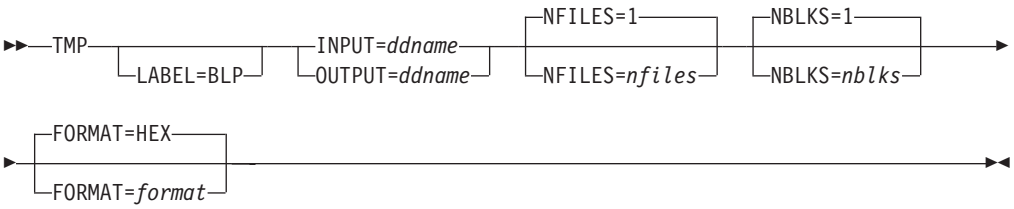
Various SET processing options allow you to control the print output:

- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Related functions

TLB Print tape labels and a tape label summary
TP Print a specified number of blocks from one or more files on a tape

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

format

The format of the output:

HEX Hexadecimal format (the default)
CHAR
Character format.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nblks Number of tape blocks to be printed from each file. The range is 0 to 99 999 999; the default is 1.

nfiles Number of tape files to be summarized. The maximum is 99 999; the default is 1.

EOV Summarizes files until a double tape mark or an EOV label is reached.
EOD Summarizes files until the delimiter specified in the EOD

parameter of the SET function is reached. The EOD delimiter is detected only if it appears in the block or blocks specified by *nblks*.

```
//TMP JOB (acct),'name'  Tape Map
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//TAPETMP  DD UNIT=561,VOL=SER=FMO001,
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM TMP INPUT=TAPETMP,NFILES=EOV
$$FILEM EOJ
/*
```

TP (Tape Print)

Purpose

Print tape files.

Usage notes

Use this function to print tape records in character format or one of the two dump formats. Printing starts with the record following the current tape position and includes any leading tape mark. Processing stops after the specified number of files, or after printing the specified number of records in any one file.

Specify a DDNAME for the tape.

For further information on the use of this function, see "Tape Print (option 4.5)" on page 357.

Options

You can specify the record format and the record length of the input data. In the case of fixed-blocked format, the block length does not need to be a multiple of the specified record length. The remainder is printed as a short record.

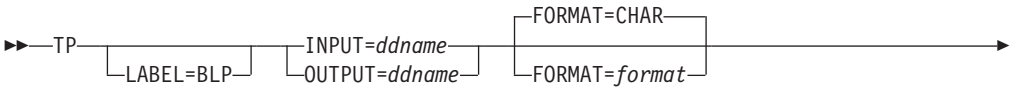
Various SET processing options allow you to control the print output:

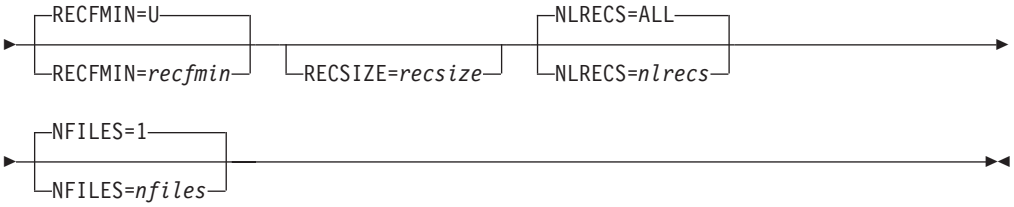
- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Related function

TMP Print a summary of the tape contents

Syntax





ddname
Refers to a DD or TSO ALLOC statement.

format
The format of the output:
CHAR
Character format (the default)
HEX Hexadecimal format.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nfiles Number of tape files to be printed. The maximum is 99 999; the default is 1.
EOV Prints files until a double tape mark or an EOV label is reached.
EOD Prints files until the delimiter specified in the EOD parameter of the SET function is reached.

If the maximum number of records per file (*nlrecs*) is reached, printing stops and the function ends regardless of the *nfiles* value.

nlrecs Maximum number of records to be printed from any one file. If this number is exceeded, printing stops. This parameter provides protection against printing a file that is much larger than expected. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records in the file, up to and including the next tape mark, are printed.

recfmin
Record format for the input. Each value is a combination of the following letters:
B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize
Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

TPEXP (Export template to XML)

Purpose
Export one or more templates to an XML format.

Usage Notes

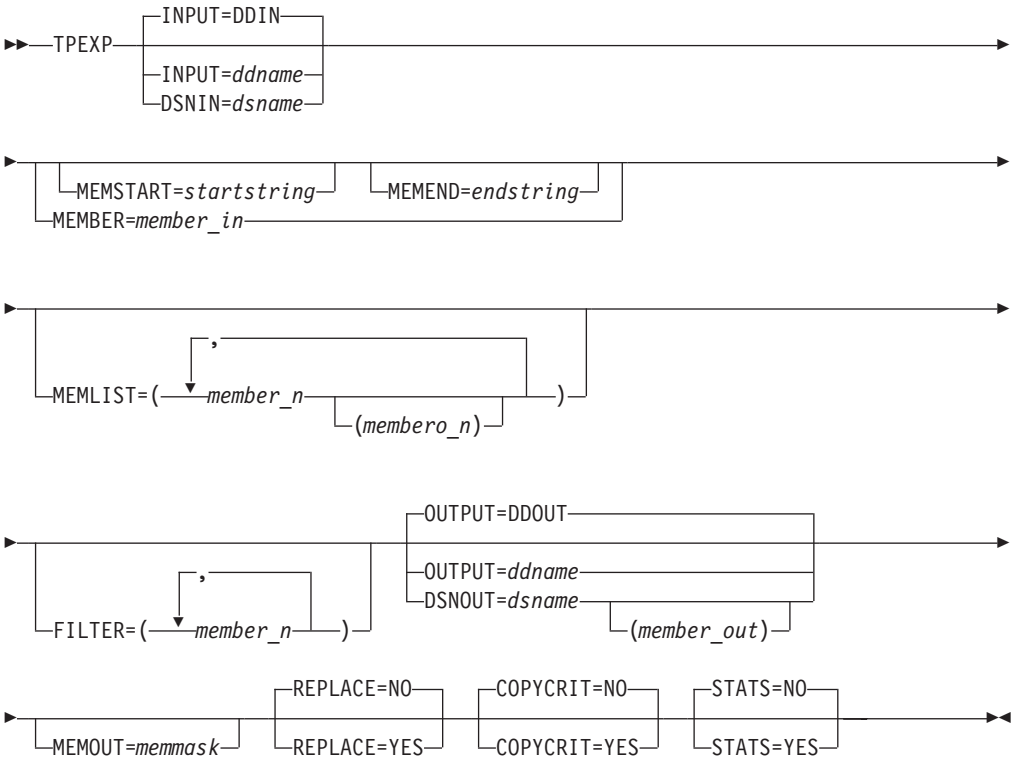
You can filter the templates selected for processing by providing a filter member names or member masks. This ensures for base or IMS templates built using copybooks that only templates that reference matching copybooks are selected for processing. For IMS views and criteria sets the filter is used to match the originating template name. Any other type of template is skipped if you provide a filter.

Related functions

TPIMP

Template import

Syntax



INPUT=ddname

This points to the input DD name which can have one or more associated data sets that must be valid template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets. You can specify concatenated data sets.

DSNIN=dsname

Data set name where templates reside.

MEMBER=member_in

The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed. A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if

you enter a member name pattern of *d*, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the PDS whose name is four characters in length are processed. member_in is ignored if the data set is not a PDS.

MEMSTART=startstring
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the startstring value onwards are included. startstring can have the same values, including wild cards, as for the member-in parameter of the MEMBER keyword.

MEMEND=endstring
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the endstring value onwards are included. endstring can have the same values, including wild cards, as for the member_in parameter of the MEMBER keyword.

FILTER
Allows you to specify a list of member names or patterns used to filter so that only templates referencing those copybooks, or copybooks that match the patterns for base and IMS templates built using copybooks. For IMS views and criteria sets the filter is taken to represent an originating template name. Any other type of template is skipped if a filter is specified.

MEMLIST
Allows you to specify a list of member names with optional associated output XML template names. If you do not specify the output member name File Manager uses the input name or the name as identified by the MEMOUT mask, memmask.

member_n
The name of the member to be processed. Generic name masks are allowed.

membero_n
Exported member name If unspecified, the exported member is not renamed.

OUTPUT=ddname
Identifies the DD card which points to the import template data set in which the exported template is/are stored or replaced. It must refer to a PDS, PDSE, or sequential data set.

DSNOUT=dsname
Defines the data set name where exported templates are created/replaced. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows: (member-out) Where DSNOUT=dsname specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=memmask
Where a number of input members have been specified, you can specify a member name pattern for the exported templates, allowing you to rename

your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter ABC* the renamed members all begin with "ABC" followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter "%%%A*" the first three characters of the renamed members remain unchanged, the fourth character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.
NO Like-named XML templates are not replaced.
YES Like-named XML templates in the output partitioned data set are replaced.

COPYCRIT
Specifies whether or not to limit the output XML to copybook definitions and criteria. This option is ignored when processing dynamic template as <symbol> elements are required to import a dynamic template.
NO All XML elements describing the template are produced.
YES XML is limited to the elements required to create the template and criteria.

STATS
Specifies ISPF member statistics are to be either created or updated by the export process.
NO Do not update or create ISPF statistics for the exported member(s).
YES ISPF statistics are updated or created for the exported member(s).

Example: Base export of all templates with member names starting with D through to members starting with C. The exported members are renamed to start with X.

```
//FMBAT EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPEXP DSNIN=FMN.TEMPLATE.EXAMPLE,
$$FILEM MEMSTART=D*,
$$FILEM MEMEND=C*,
$$FILEM MEMOUT=X*,
$$FILEM DSNOUT=FMN.XML.TEMPLATE,
$$FILEM COPYCRIT=YES,
$$FILEM STATS=YES,
$$FILEM REPLACE=YES
```

It produces this report:

```
          Template Export Report
Template      New name   Type       Status
-------------------------------------------------------------
```

```
DJ1E          XJ1E       IMS        Exported
DJ1ECR2       XJ1ECR2    IMS CRIT   Exported
DJ1EVW        XJ1EVW     IMS VIEW   Exported
CTEMP1        XTEMP1     BASE       Exported
CTEMPDYN      XTEMPDYN   DYNAMIC    Exported
CTEMP3        XTEMP3     BASE       Exported
FMNBA631 6 members read 6 Exported 0 Export replaced  0 Errors
```

Table 15. Export status and action

Status	Explanation	Action
Exported	Template successfully exported to XML format.	None
Exported replaced	Template successfully exported to XML format and existing member replaced	None
Not replaced	Output member exists and REPLACE=NO is specified.	If you want to replace the member specify REPLACE=YES
Corrupt template	The input template could not be loaded because it is corrupt.	Recreate the input template and rerun the job.
Template allocate error	The input template data set could not be allocated, possibly not found.	Correct the input data set name and rerun the job.
Not found	The input member name could not be found on the input template data ser.	Correct the member name or data set name and rerun the job.
Unexpected error	There should be previous error messages describing the error that occurred.	Look at the preceding error messages and address accordingly

TPIMP (Import template from XML)

Purpose

Import one or more templates from an XML format.

Usage Notes

The executed program to import an IMS template, view or criteria set must be FMNIMS. The executed program to import a DB2 template must be FMNDB2. For DB2 templates you must specify the SSID on the execute

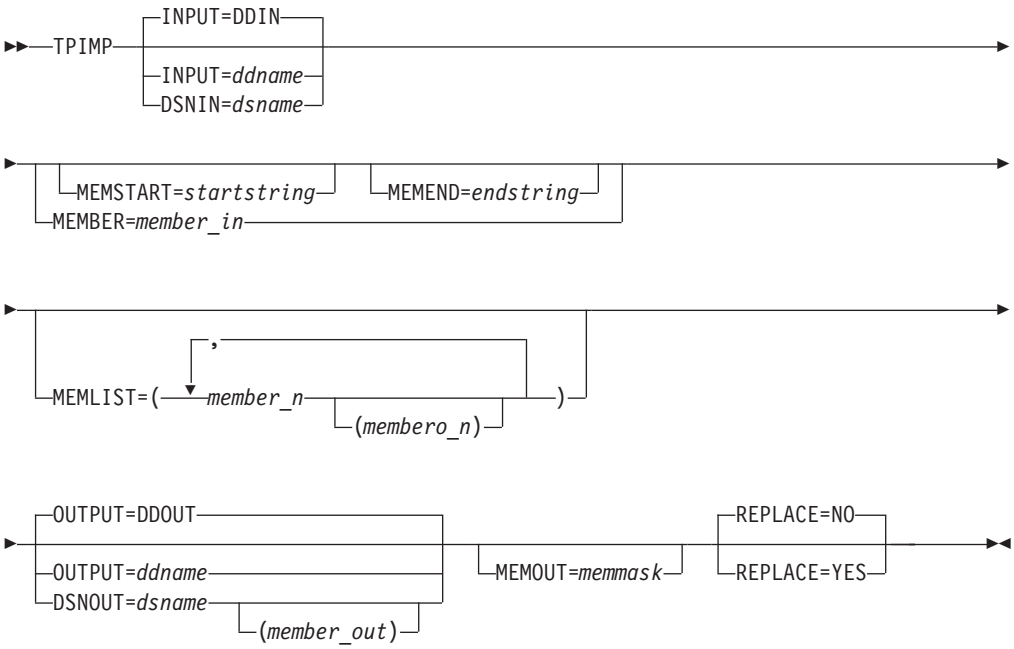
PARM, and the DB2 object referred to in the XML <db2object> element must exist in the DB2 subsystem for the import to be successful.

Related functions

TPEXP

Template export

Syntax



INPUT=ddname

This points to the input DD name which can have one or more associated data sets that must be valid XML template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets. You can specify concatenated data sets.

DSNIN=dsname

Data set name where templates reside.

MEMBER=member_in

The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed. A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

***** represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of *d*, all members in the PDS whose name contains "d" are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of %%%%, all members in the PDS whose name is four characters in length are processed. member_in is ignored if the data set is not a PDS.

MEMSTART=startstring
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the startstring value onwards are included. startstring can have the same values, including wild cards, as for the member-in parameter of the MEMBER keyword.

MEMEND=endstring
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the endstring value onwards are included. endstring can have the same values, including wild cards, as for the member_in parameter of the MEMBER keyword.

FILTER
Allows you to specify a list of member names or patterns used to filter so that only templates referencing those copybooks, or copybooks that match the patterns for base and IMS templates built using copybooks. For IMS views and criteria sets the filter is taken to represent an originating template name. Any other type of template is skipped if a filter is specified.

MEMLIST
Allows you to specify a list of member names with optional associated output template names. If you do not specify the output member name File Manager uses the input name or the name as identified by the MEMOUT mask, memmask.

member_n
The name of the member to be processed. Generic name masks are allowed.

membero_n
Imported member name If unspecified, the imported member is not renamed.

OUTPUT=ddname
Identifies the DD card which points to the import template data set in which the imported template is/are stored or replaced. It must refer to a PDS, PDSE, or sequential data set.

DSNOUT=dsname
Defines the data set name where imported templates are created/updated. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows: (member-out) Where DSNOUT=dsname specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=memmask
Where a number of input members have been specified, you can specify a member name pattern for the imported templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent

signs. For example, if you enter ABC* the renamed members all begin with "ABC" followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter %%%A* the first three characters of the renamed members remain unchanged, the fourth character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates are updated with the XML definitions.

YES Like-named templates in the output partitioned data set are replaced.

Example 1. Base import of all template with member names beginning with a and renaming the output member names to all begin with Z.

```
//FMBAT EXEC PGM=FILEMGR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$FILEM TPIMP DSNIN=FMN.XML.EXAMPLE,
$$FILEM MEMBER=A*,
$$FILEM MEMOUT=X*,
$$FILEM DSNOUT=FMN.IMP.EXAMPLE,
$$FILEM REPLACE=YES
```

It produces this report:

```
          Template Import Report
Template      New name   Type       Status
-----------------------------------------------------------------
ABEND         XBEND      BASE       Import replaced
AODA0140      XODA0140   DYNAMIC    Import replaced
ASMTEST       XSMTEST    BASE       Import replaced
FMNBA635 3 members read 0 Imported 3 Import replaced  0 Errors
```

Example 2. Import an IMS template, criteria set and a view specifying member names

```
//FMBAT EXEC PGM=FMNIMS
//SYSPRINT DD SYSOUT=*
//FMIMSIN DD *
$$FILEM SET HEADERPG=YES,PAGESIZE=60
$$FILEM TPIMP DSNIN=FMN.XML.EXAMPLE,
$$FILEM MEMLIST=(DJ1E,
$$FILEM          DJ1ECR2,
$$FILEM          DJ1EVW),
$$FILEM DSNOUT=FMN.IMP.EXAMPLE,
$$FILEM REPLACE=YES
```

It produces this report:

```
          Template Import Report
Template      New name   Type       Status
-------------------------------------------------------------
DJ1E                     IMS        Import replaced
DJ1ECR2                  IMS CRIT   Imported
DJ1EVW                   IMS VIEW   Imported
FMNBA635 3 members read 2 Imported 1 Import replaced  0 Errors
```

Table 16. Import status and action

Status	Explanation	Action
Imported	Template successfully imported from XML format.	None
Import replaced	Template successfully imported from XML format and existing member replaced	None
Import updated	Template successfully imported from XML format and existing member updated.	None.
Not found	The input member name could not be found on the input template data ser.	Correct the member name or data set name and rerun the job.
Unexpected error	There should be previous error messages describing the error that occurred.	These are normally errors in the XML definitions. The error messages should indicate the failing line and reasons. Correct the XML and rerun the job. (See appendix nn for XML reference.

TRS (Tape Record Scan)

Purpose

Locate data within a tape file.

Usage notes

You can use this function to locate tape records containing data that meet your search conditions. Scanning starts at the current tape position. Scanning applies to logical records, if a record size or a variable record format is indicated.

The search string can contain up to 50 characters and can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in quotation marks. The string can contain blanks and commas and must match exactly (including case).
- A character string enclosed in quotation marks. The string can contain blanks and commas and must match exactly (including case).
- A hexadecimal string, for example X'04'.

Specify a DDNAME for the tape.

For further information on the use of this function, see “Tape Record Scan (option 4.10)” on page 363.

Options

Specify a record format or a length to deblock the physical records before searching. You can request searching everywhere in the data, from a position you specify or only at the specified offset. You can also specify the number of blocks to be scanned and the number of occurrences of the string to be found before the scan stops. The scan stops when the specified number of blocks is reached or when the specified number of string occurrences have been found, whichever occurs first.

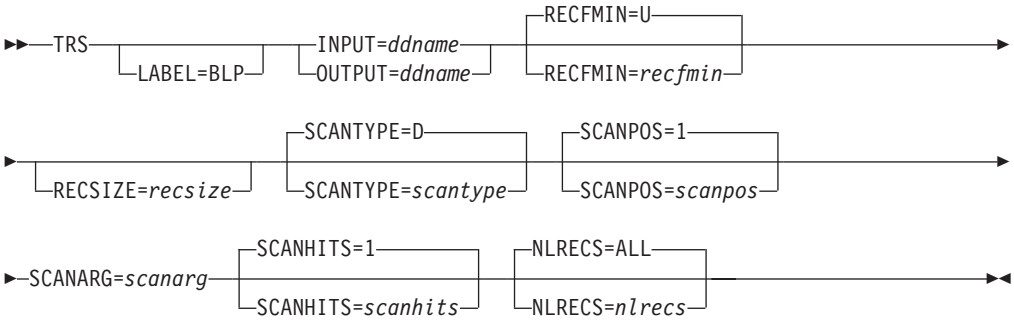
Various SET processing options allow you to control the print output:

- SET PRINTOUT defines the destination of the print output. If set to PRINTOUT=SYSOUT, you can use the PB (Print Browse) function to browse the accumulated output.
- SET RECLIMIT controls how many bytes of each record are printed.
- The format of the print output also depends on the settings of SET PAGESIZE, SET PRINTLEN, and SET PRTTRANS.
- Use SET DUMP to specify the dump format.

Related functions

FCH Find or change data
TP Print a specified number of blocks from one or more files on a tape

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nlrecs Number of records to be scanned or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, File Manager scans to the end of the file or until the specified number of hits is found.

recfmin

Record format for the input. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize
Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

scanarg
The string to be searched for. The string can contain up to 50 characters and can be:
- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
- X followed by a hexadecimal string enclosed in quotation marks (for example, X'C1C2C3').

scanhits
Specifies whether scanning stops after the first match is found, after a specified number of matches, or at the end of the file. The maximum is 99 999 999; the default is 1. Specify ALL to scan to the end of the file.

scanpos
Offset (byte position) where the search begins within a record. The default is 1.

scantype
One of the following:

D Scans records *only* at the position in the record specified by *scanpos* (the default)

DP Scans records *starting* at the position specified by *scanpos*

```
//TRS JOB (acct),'name'  Tape Record Scan
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=561,VOL=SER=FMO001,
//            DISP=SHR,LABEL=(,BLP)
//SYSIN    DD *
$$FILEM TRS INPUT=TAPE,SCANARG='HDR2'
$$FILEM EOJ
/*
```

TS (Tape to Sequential Data)

Purpose
Copy tape records to a sequential data set.

Usage notes
Use this function to copy all or selected data from a tape file to a sequential data set on disk, on tape, or SYSOUT. You can also use this function to convert a multiple file tape into a single sequential data set (tape image file), which can be distributed across a network. For more information, see “Copying tape data across a network” on page 354. Specify a number of files other than one, but no tape file ID or start position.

File Manager converts tape marks into special records containing "FILEMTMFILEMTM...". When restoring the file to tape using the ST (Sequential Data to Tape) function, File Manager converts the special records back into tape marks. The resulting tape is an exact copy of the original tape.

Specify a DDNAME for the tape.

If you want to use allocation parameters that are not supported by File Manager, you must use the TSO ALLOCATE command to allocate the data set out of File Manager with the file name QSAMOUT.

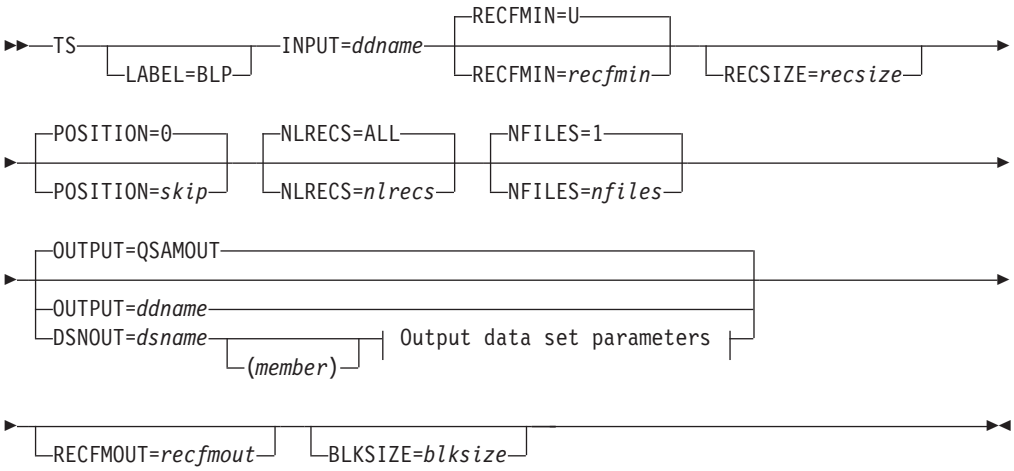
Options

You can specify the record format and the record length of the input data. You can also specify the record format and the block size of the output, and the number of records to copy.

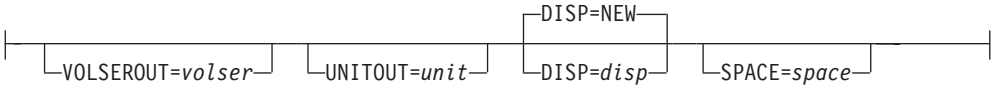
Related functions

ST Copy a data set to tape
TV Copy tape data to a VSAM data set

Syntax



Output data set parameters:



blksize

The maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.

ddname

Refers to a DD or TSO ALLOC statement. The default is QSAMOUT for output.

disp Disposition of the output data set. Specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.

Note: SMS might modify the allocation of new data sets on your system. For details, contact your SMS Administrator.

dsname
Defines the name of the data set. If specified, any DD statement provided are not used. The name may include a member name in parenthesis. If the member is specified here, the associated Member parameter must be empty. To further describe the data set, use the appropriate keywords as shown in the syntax diagram.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

member
For a PDS, the member name.

nfiles Number of tape files to be copied. The maximum is 99 999; the default is 1.

EOV Copies files until a double tape mark or an EOV label is reached.
EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.

Use this parameter to copy a multifile tape, including tape marks, to a single sequential data set. (You can use the ST function to copy the data set back to tape, including the tape marks.) For more information, see "Copying tape data across a network" on page 354.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

recfmin
Record format for the input. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recfmout
Record format for the output. Each value is a combination of the following letters:

A Containing ISO/ANSI control characters
B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
M Containing machine code control characters
S Standard format (fixed length) or spanned format (variable length)
U Undefined length
V Variable length

The possible values are: U, UA, UM, F, FA, FM, FB, FBA, FBM, FBS, FBSA, FBSM, V, VA, VM, VB, VBA, VBM, VS, VSA, VSM, VBS, VBSA, VBSM, D, DA, DB, DBA, DS, DSA, DBS, and DBSA.

Use *recfmout* to override existing DCB information. You cannot create a PDS with a record format of FBS. If output data sets with a record format containing the letter D are written to disk, unpredictable results might occur.

recsize
Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

skip Number of logical records to be skipped. The default is 0.

space The number of space units to be allocated for a new data set. Specify *n* or (*n*,*m*), where *n* is the number of primary units, and *m* is the number of secondary units. The space unit is a track.

tapevol
Serial number (external label or VOL1) of the tape volume to be mounted.

unit A device number. A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

volser Volume serial number for a new or non-cataloged data set.

```
//TS  JOB (acct),'name'  Create Tape Image File
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FILEMGR  EXEC FMBAT
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=(561,,DEFER),VOL=SER=FMO001,
//            DISP=SHR,LABEL=(,BLP)
//XMIT     DD DSN=FMNUSER.TAPE.XMIT.WORK,
//            DISP=(,CATLG),
//            UNIT=SYSALLDA,
//            SPACE=(CYL,(1,1),RLSE),
//            VOL=SER=MVS1T3
//SYSIN    DD *
$$FILEM TS  INPUT=TAPE,LABEL=BLP,
$$FILEM     OUTPUT=XMIT,NFILES=EOV
$$FILEM EOJ
/*
```

TT (Tape to Tape)

Purpose
Copy tape data from one tape to another.

Usage notes
Use this function to copy tape files from one tape to another for an exact copy. No label processing is involved. Label sets are copied as files.

For each tape file copied, block count statistics are printed or displayed. Records are always copied as physical blocks, regardless of whether they are labels or ordinary data. Any leading tape mark is copied.

Specify a DDNAME for the tape.

Unless you have strong reasons for not doing so, use bypass label processing (BLP) with this function.

For further information on tape copy functions, see "Tape to Tape (option 4.2.1)" on page 350.

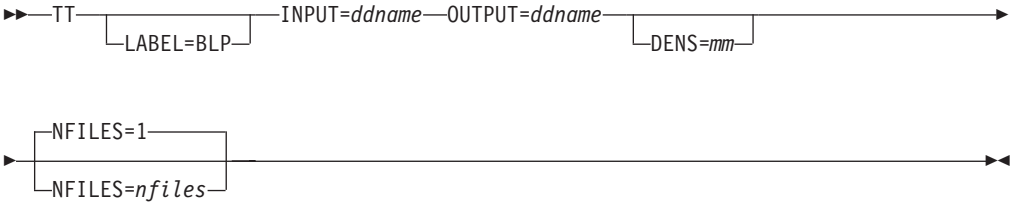
Options

You can specify the number of tape files to be copied, or take the default of one. Copying starts at the current position and stops after the specified number of files.

Related functions

DSC Copy a data set to another one
TLT Copy multivolume, multiple-file labeled tapes
TTC Compare two tapes byte by byte
TTR Copy a file from one tape to another with reblocking

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

nfiles Number of tape files to be copied. The maximum is 99 999; the default is 1.

EOV Copies files until a double tape mark or an EOV label is reached.
EOT Copies files until the physical end of the volume
EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.

The records from the current tape position up to and including the next tape mark make up one file. However, if the tape is positioned immediately before a tape mark, the file is considered to end at the next tape mark.

```
//TT  JOB (acct),'name'  Tape to Tape Copy
//FILEMGR   EXEC PGM=FILEMGR
//SYSPRINT  DD SYSOUT=*
//TAPE1     DD UNIT=381,VOL=SER=FM01,
//             DISP=OLD,LABEL=(,BLP)
//TAPE2     DD UNIT=382,VOL=SER=FM02,
//             DISP=OLD,LABEL=(,BLP)
//SYSIN     DD *
```

```
$$FILEM TT  INPUT=TAPE1,OUTPUT=TAPE2,
$$FILEM     NFILES=EOV
$$FILEM EOJ
/*
```

TTC (Tape to Tape Compare)

Purpose

Compares two tapes byte by byte.

Usage notes

Use this function to compare, byte by byte, the data on one tape with the data on another. This function is particularly useful if you have made a copy of a tape, and you want to check that the duplicate is exactly the same as the original. Comparison starts at the current tape position of each tape. When the first difference is found, comparison stops and the comparison results are printed.

Specify a DDNAME for the tape.

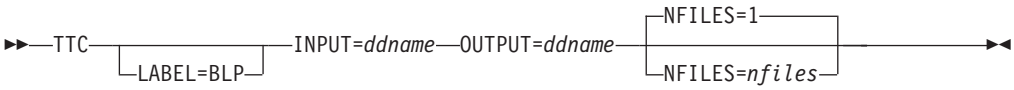
Options

You can compare the entire tape, or a specific number of files.

Related functions

TLT Copy multivolume, multiple-file labeled tapes
TT Copy files from one tape to another without label processing
TTC Compare two tapes byte by byte
TTR Copy a file from one tape to another with reblocking

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *File Manager Customization Guide*.

nfiles Number of tape files to be compared. The maximum is 99 999; the default is 1.

EOV Compares files until a double tape mark or an EOV label is reached.
EOD Compares files until the delimiter specified in the EOD parameter of the SET function is reached.

```
//TTC JOB (acct),'name'  Tape to Tape Compare
//FILEMGR   EXEC PGM=FILEMGR
//SYSPRINT  DD SYSOUT=*
//TAPE1     DD UNIT=381,VOL=SER=FM01,
//             DISP=OLD,LABEL=(,BLP)
//TAPE2     DD UNIT=382,VOL=SER=FM02,
//             DISP=OLD,LABEL=(,BLP)
//SYSIN     DD *
```

```
$$FILEM TTC INPUT=TAPE1,OUTPUT=TAPE2,
$$FILEM     NFILES=EOV
$$FILEM EOJ
/*
```

TTR (Tape to Tape Reblocked)

Purpose

Copy one file from tape to tape with reblocking.

Usage notes

Use this function to copy one tape file, starting from the current tape position, to one or more output tapes. You can change the record format and reblock the file during copying. Any leading tape mark is copied. No label processing is done. Any label sets on input are copied as files.

Specify a DDNAME for the tape.

Unless you have strong reasons for not doing so, use bypass label processing (BLP) with this function.

For further information about tape copy functions, see "Tape to Tape (option 4.2.1)" on page 350.

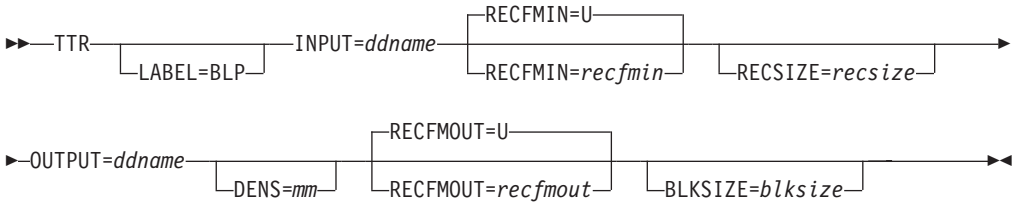
Options

You can specify the input and output record format, the input record size and the output block size.

Related functions

DSC Copy a data set to another one
TLT Copy multivolume, multiple-file labeled tapes
TT Copy files from one tape to another without label processing
TTC Compare two tapes byte by byte

Syntax



blksize

If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape is processed by other utilities or standard access methods, you must also consider the operating system limits.

ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

recfmin

Record format for the input. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recfmout

Record format for the output. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize

Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

```
//TTR JOB (acct),'name'  Tape to Tape Reblocked
//FILEMGR   EXEC PGM=FILEMGR
//SYSPRINT  DD SYSOUT=*
//TAPE1     DD UNIT=381,VOL=SER=FM01,
//             DISP=OLD,LABEL=(,BLP)
//TAPE2     DD UNIT=382,VOL=SER=FM02,
//             DISP=OLD,LABEL=(,BLP)
//SYSIN     DD *
$$FILEM TTR INPUT=TAPE1,RECFMIN=VB,
$$FILEM     OUTPUT=TAPE2,RECFMOUT=U
$$FILEM EOJ
/*
```

TV (Tape to VSAM)

Purpose

Copy tape records to a VSAM data set.

Usage notes

Use this function to copy all or selected tape records to a VSAM data set. You can also use this function to convert a multiple file tape into a single VSAM ESDS (tape image file), which can be distributed across a network. For more information, see "Copying tape data across a network" on page 354.

Specify a number of files other than one, but no tape file ID or start position. File Manager converts tape marks into special records containing "FILEMTMFILEMTM...". When restoring the file to tape using the VT (VSAM to Tape) function, File Manager converts the special records back into tape marks. The resulting tape is an exact copy of the original tape.

Specify a DDNAME for the tape.

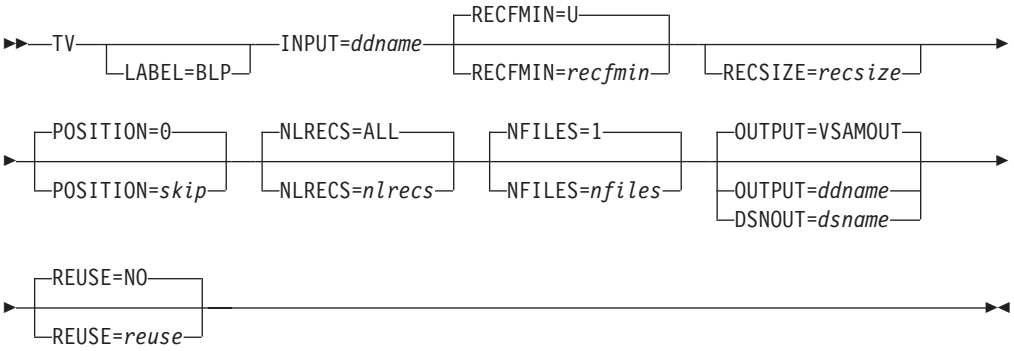
Options

You can specify the record format and the record length of the input data, and the position of the first record to copy. You can also specify the number of records to be copied, and whether any existing records in the output data set are to be replaced.

Related functions

DSC Copy a data set to another one
TS Copy tape data to a data set
VT Copy a VSAM data set to tape

Syntax



ddname

Refers to a DD or TSO ALLOC statement. The default is VSAMOUT for output.

dsname

Name of a VSAM data set.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nfiles Number of tape files to be copied. The maximum is 99 999; the default is 1.

EOV Copies files until a double tape mark or an EOV label is reached.
EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.

Use this parameter to copy a multifile tape, including tape marks, to a single VSAM entry-sequenced data set. (You can use the VT function to copy the VSAM file back to tape, including the tape marks.) For more information, see "Copying tape data across a network" on page 354.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

recfmin

Record format for the input. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records

F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize
Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

reuse Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

skip Number of logical records to be skipped. The default is 0.

```
/* REXX */
/* copy a tape file to a vsam data set       */
/* allocate input tape                       */
/* use SL processing.  required file is at   */
/* position 1                                */

"ALLOC FILE(TAPE) DA('FMNUSER.TAPEIN')",
      "VOLUME(FMO001) UNIT(CART) LABEL(SL)",
      "POSITION(1) OLD"

/* copy a tape file to a vsam data set       */

"FILEMGR $TV INPUT=TAPE,",
            "DSNOUT=VSAM.DISK.VT55.DATA"

/* free the tape unit                        */

"FREE FILE(TAPE)"

return
```

TX (Tape to REXX Variable) — REXX only

Purpose
Copy tape records into a REXX stem variable.

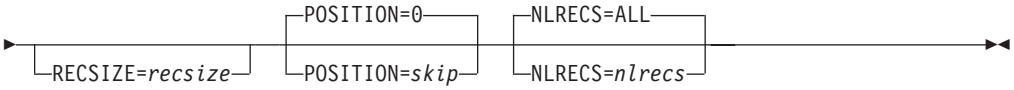
Usage notes
Each record is copied to a variable named *stem.nnn*. *stem*.0 is a counter.

For example, if you copy 3 records, *stem* is VARXX. and VARXX.0 has the value 5:
- The records are copied to VARXX.6, VARXX.7, and VARXX.8.
- The value of VARXX.0 is changed to 8.

This function is available only from a REXX procedure.

Related functions
TS Copy tape data to a data set
XT Copy a REXX stem variable to a tape file

Syntax



ddname
Refers to a DD or TSO ALLOC statement.

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

recfmin
Record format for the input. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

recsize
Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

skip Number of logical records to be skipped. The default is 0.

stem A REXX stem variable name. The maximum length is 44 characters.

```
/* REXX */
/* copy a tape file to a rexx stem          */
/* allocate input tape                      */
/* use SL processing.  required file is at  */
/* position 1                               */

"ALLOC FILE(TAPE) DA('FMNUSER.TAPEIN')",
      "VOLUME(FMO001) UNIT(CART) LABEL(SL)",
      "POSITION(1) OLD"

/* copy a tape file to a rexx stem          */

"FILEMGR $TX INPUT=TAPE,",
           "VARNAME=TAPE."


/* Show stem contents                     */

Do i=1 To tape.0
  Say tape.i      /* show tape record       */
  End

/* free the tape unit                       */

"FREE FILE(TAPE)"

return
```

VER (Display Service Level)

Purpose
Display the current level of File Manager you are using.

Usage notes
The VER function also displays the latest PTF applied.

Options
None.

No parameters.

VLM (View Load Module)

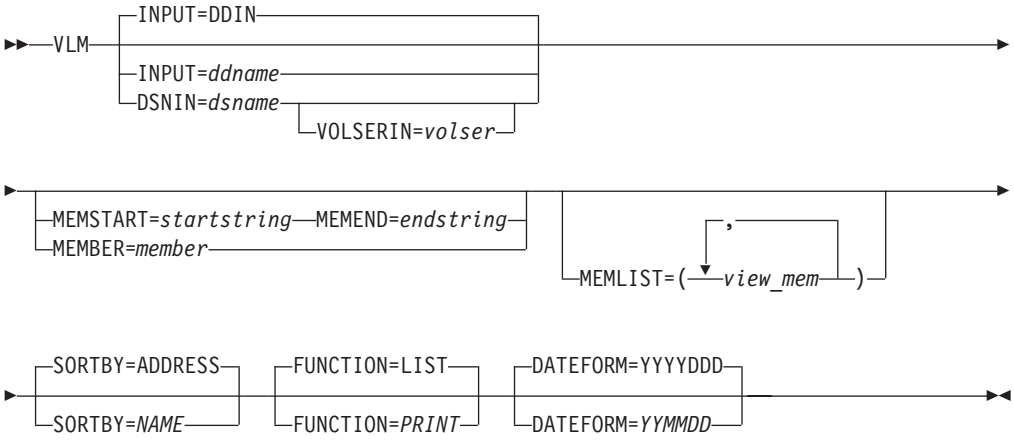
Purpose
List symbols in a load module.

Usage notes
Use this function to print a list of the symbols (CSECTs, common sections, entry points, and ZAPs) in a load module.

Options
The list can be sorted by symbol address (offset within module) or symbol name.

Related functions
None.

Syntax



INPUT=*ddname*
Refers to a DD or TSO ALLOC statement. The default is DDIN for input, and DDOUT for output.

DSNIN=*dsname*
Defines the name of the input data set. If specified, any DD statement provided are not used. The name may include a member name or pattern in parenthesis. If the member is specified here, the associated Member parameter must be empty. To further describe the data set, use the appropriate keywords as shown in the syntax diagram. You can further describe this data set using:

VOLSER=*volser*
Volume serial number for a non-cataloged data set.

MEMBER=*member*
The name of a single member or pattern in a PDS load module.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included in the processing. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included in the processing. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *memold* parameter of the MEMOLD keyword.

MEMLIST
Provides a means of selecting members from a load library where no generic name pattern and no member name range has been specified. If the MEMLIST keyword is specified, only those members included in the MEMLIST arguments are processed. Members selected by the MEMBER=*member* that are not included in the MEMLIST arguments are not processed.

view_mem
The name of the member to be processed. Generic name masks are not allowed.

FUNCTION
The command that you want to execute:
LIST Display the list (the default)
PRINT
Print the list

SORTBY
Sorts the list or print output by one of the following:
ADDRESS
Sort by the address of the symbol
NAME
Sort by the name of the symbol

DATEFORM
Specifies Gregorian or Julian format of dates:
YYYYDDD
Dates shown in the YYYY.DDD format
YYMMDD
Dates shown in YY/MM/DD format

Note: The function parameter is ignored in batch mode.

```
//VLM JOB (acct),'name' View Load Module
//*
//FMBAT    PROC
//FMBAT    EXEC PGM=FMNMAIN
//STEPLIB  DD DSN=FMNUSER.FMN310.TSTLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//FMNMAIN EXEC FMBAT
```

```
//DDIN     DD DSN=FMNUSER.FMN310.TSTLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$$FILEM VLM MEMBER=FMNMAIN
$$FILEM EOJ
/*
```

VO (VSAM to Object)

Purpose

Copy data from a VSAM data set to one or more OAM objects.

Usage notes

Use this function to copy data from a VSAM data set to an object database. If the input is a backup data set created with the OV (Object to VSAM) function, the object header records are used for restoring.

If loading from an object backup data set the objects are restored with the original name into the original collection. To filter loading, specify a collection name, a generic object name, the location (DASD or OPTICAL, or a specific optical volume), and a creation date range. The filter applies to the object header information saved with the objects.

If the input does not contain backup copies of multiple objects, one object is created. You can specify a collection name and an object name. If the input data begins with an object header record, the default values are taken from it. The object is made up of the concatenated input records, excluding any object header.

For more information about using this function, see “Restoring OAM objects” on page 388.

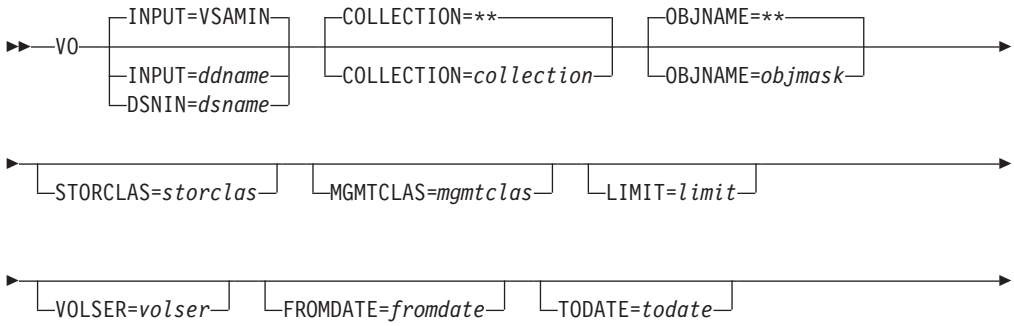
Options

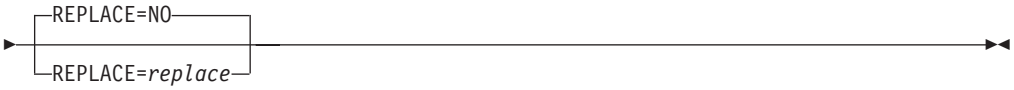
You can specify SMS storage class and management class values to be used for storing the object; these might, however, be overridden by your installation defaults. You can also specify whether an existing object with matching name is to be replaced in the output collection.

Related functions

OO Copy an object to the same or another collection
OV Backup objects from an OAM database to a VSAM data set
SO Copy a data set to an object database

Syntax





collection
If the data set contains one object with a header, you can optionally specify a collection name to override the collection name in the header.

If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify the collection where the object is stored.

If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify a collection name to restore only objects from the specified collection.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

ddname
Refers to a DD or TSO ALLOC statement. The default is VSAMIN.

dsname
Name of a VSAM data set.

fromdate
The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.

limit To limit the objects that are copied based on where the objects were located (before they were backed up to a data set), specify one of the following:

DASD
Only objects that were stored on DASD

OPTICAL
Only objects that were stored on optical disks

VOLUME
Only objects that were stored on a specified optical disk. You must also specify the volser of the optical disk.

mgmtclas
Name of an SMS management class.

objmask
If the data set contains one object with a header, you can optionally specify an object name to override the object name in the header.

If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify a name for the object.

If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify an object name or generic object name to limit the objects that are restored based on object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) with any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

replace
Specify YES to replace any existing objects with the same name in the collection. By default, the object is not copied.

storclas
Name of an SMS storage class.

todate The latest creation date, in the form *yyyymmdd*.

volser If you specified VOLUME for *limit*, the volume serial number of the optical disk.

VT (VSAM to Tape)

Purpose
Copy VSAM records to a tape file.

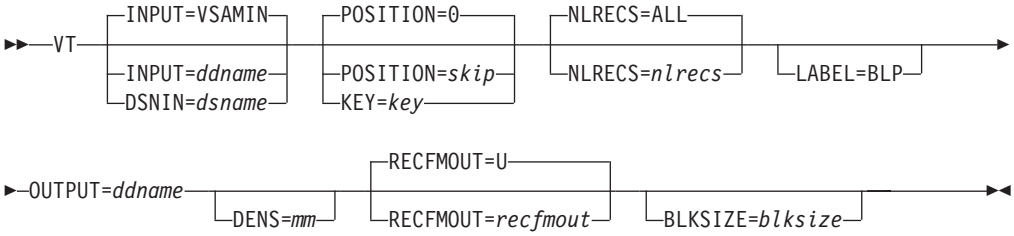
Usage notes
If you used the TV (Tape to VSAM) function to copy a multifile tape to a single VSAM ESDS (tape image file), you can use VT to copy the file back to tape. File Manager converts the special records back into tape marks. The resulting tape is an exact copy of the original tape.

Specify a DDNAME for the tape.

Options
You can specify the number of records to skip from the beginning of the input data set, or the key or slot value to start processing. You can also specify the output record format and block size.

Related functions
DSC Copy a data set to another one
TV Copy tape data to a VSAM data set
VO Copy VSAM data to an object database

Syntax



blksize
If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape is processed by other utilities or standard access methods, you must also consider the operating system limits.

ddname
Refers to a DD or TSO ALLOC statement. The default is VSAMIN for input. To create the output tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

dsname
Name of a VSAM data set.

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

LABEL=BLP
Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

recfmout
Record format for the output. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

skip Number of logical records to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, copying begins with the first record in the data set.

```
//VT  JOB (acct),'name'  VSAM to TAPE
//FMBAT    PROC
//FMBAT    EXEC PGM=FILEMGR
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//IDCPROC  PROC
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//CREATE   EXEC IDCPROC,REGION=6M
//SYSIN    DD *
 DELETE FMNUSER.VSAM.VT55 CLUSTER PURGE ERASE
 DEFINE CLUSTER ( -
      NAME( FMNUSER.VSAM.VT55 ) -
      CISZ(4096) -
      FREESPACE(30 30) -
      NONINDEXED -
      KEYS(12 0) -
      RECORDS(1000 100) -
```

```
      RECORDSIZE(133 133) -
      REUSE -
      VOL(MVS1W2) ) -
    DATA( NAME(FMNUSER.VSAM.VT55.DATA) )
/*
//*
//STP1005  EXEC FMBAT
//VSAMIN   DD DISP=SHR,DSN=FMNUSER.VSAM.VT55
//TAPE     DD UNIT=CART,
//            VOL=(,RETAIN,,,SER=FMO001),
//            LABEL=(1,SL),DISP=(,KEEP),
//            DSN=FMNUSER.TAPEOUT
//SYSIN    DD *
$$FILEM VER
*           GENERATE SOME DATA AND PLACE
*           IN A VSAM FILE
$$FILEM DSG OUTPUT=VSAMIN,REUSE=NO,NLRECS=4000,
$$FILEM     RECSIZE=133,FILLCHAR=BIN,KEYLOC=1
*           NOW COPY THE VSAM FILE TO TAPE
$$FILEM VT  INPUT=VSAMIN,OUTPUT=TAPE
$$FILEM EOJ
/*
```

VX (VSAM to REXX Variable) — REXX only

Purpose

Copy records from a VSAM data set into a REXX stem variable.

Usage notes

Each record is copied to a variable named *stem.nnn*. *stem*.0 is a counter.

For example, if you copy 3 records, *stem* is VARXX. and VARXX.0 has the value 5:

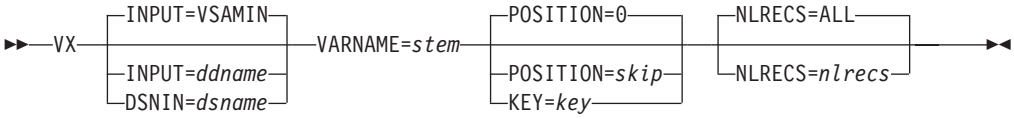
- The records are copied to VARXX.6, VARXX.7, and VARXX.8.
- The value of VARXX.0 is reset to 8.

This function is available only from a REXX procedure.

Related function

XV Copy a REXX stem variable to a VSAM data set

Syntax



ddname

Refers to a DD or TSO ALLOC statement. The default is VSAMIN.

dsname

Name of a VSAM data set.

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in quotation marks. You can also specify a key in hexadecimal format (for example, X'C1C2C3').

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

skip Number of logical records to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, copying begins with the first record in the data set.

stem A REXX stem variable name. The maximum length is 44 characters.

```
/* REXX */
/* Copy a VSAM data set to a REXX stem     */
/* Change input.vsam.file to the name      */
/* of the required VSAM file               */

"FILEMGR $VX  DSNIN='input.vsam.file',",
            "VARNAME=STEM."

/* Show stem contents                      */

do i = 1 to stem.0;
    say 'Record' i '=' stem.i;
    end;

return;
```

WTM (Write Tape Mark)

Purpose

Write one or more tape marks.

Usage notes

Tape marks are special records on tape. Data between two tape marks make up a tape file. Two consecutive tape marks are considered a null file or indicate the end of data on a tape volume (if after an end-of-file or an end-of-volume label, or on a NL tape).

Specify a DDNAME for the tape.

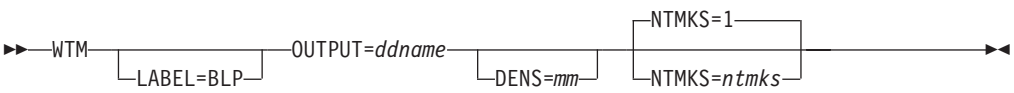
Options

None.

Related function

INT Initialize a tape

Syntax



ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

ntmks Number of tape marks to be written. The maximum is 9 999; the default is 1.

XT (REXX Variable to Tape) — REXX only

Purpose

Copy a REXX stem variable to a tape file.

Usage notes

Each variable is copied, starting with *stem*.1 and finishing with *stem.nnn*, where *nnn* is the value of *stem*.0.

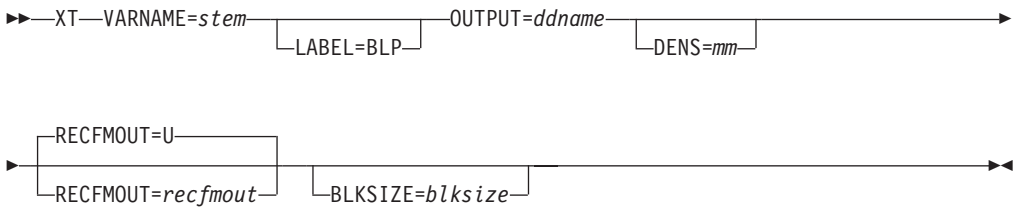
For example, if *stem* is VARXX. and VARXX.0 has the value 3, the variables VARXX.1, VARXX.2, and VARXX.3 are copied to a tape file.

This function is available only from a REXX procedure.

Related function

TX Copy tape records into a REXX stem variable

Syntax



blksize

If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape is processed by other utilities or standard access methods, you must also consider the operating system limits.

ddname

Refers to a DD or TSO ALLOC statement.

LABEL=BLP

Specifies that bypass label processing is used. This parameter must be specified with the first File Manager function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the *File Manager Customization Guide*.

mm A 2-byte tape mode as shown in "Tape density and mode values" on page 340.

recfmout

Record format for the output. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

The possible values are: U, F, FB, V, VB, VBS, VS, D, DB, DBS, and DS.

stem A REXX stem variable name. The maximum length is 44 characters.

XV (REXX Variable to VSAM) — REXX only

Purpose

Copy a REXX stem variable to a VSAM data set.

Usage notes

Each variable is copied, starting with *stem*.1 and finishing with *stem.nnn*, where *nnn* is the value of *stem*.0.

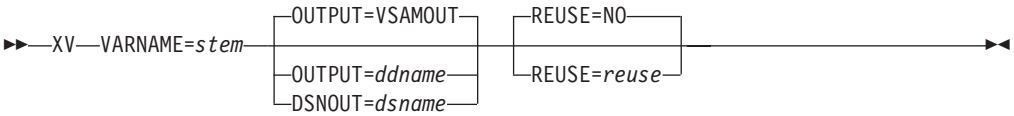
For example, if *stem* is VARXX. and VARXX.0 has the value 3, the variables VARXX.1, VARXX.2, and VARXX.3 are copied to a VSAM data set.

This procedure is available only from a REXX procedure.

Related function

VX Copy records from a VSAM data set to a REXX stem variable

Syntax



ddname

Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.

dsname

Name of a VSAM data set.

reuse Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

stem A REXX stem variable name. The maximum length is 44 characters.

External REXX functions

This section describes the REXX external functions that you can use when writing REXX procedures to enhance a File Manager function. In addition, File Manager provides access to the REXX internal functions described in the *z/OS TSO/E REXX Reference*.

CHANGE

Changes a specified character string.

CHG_OUT

(Can be used in FASTREXX procedures.) Changes a character string in the output record.

CHG_VAR

(Can be used in FASTREXX procedures.) Changes one or more occurrences of an old string in a variable to a new string.

CONTAINS

Checks for character values in a specified string.

FLD (Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Refers to a field from the current input record.

FLD_CO
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Searches a field in the input record for one or more occurrences of a string, or tests a field in the input record for one more numeric values.

FLD_OUT
(Can be used in FASTREXX procedures.) Overlays the output record with a field from the input record.

FLD_TM
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Tests selected bits of a field in the input record.

FLD_TYPE
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Tests the data type of a field in the input record.

FLDI (Can be used in FASTREXX procedures.) Performs a conditional test against an input record field.

FLDO (Can be used in FASTREXX procedures.) Performs a conditional test against an output record field.

I_LENGTH
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Returns the length of the input record.

MOD_DATE
(Can be used in FASTREXX procedures.) Sets, increments, or decrements a date field using year, month, or day values.

NCONTAIN
Checks for numeric values in a specified string.

O_LENGTH
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Returns the current length of the output record.

OFLD_CO
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Searches a field in the output record for one or more occurrences of a string, or tests a field in the output record for one more numeric values, and resets the current output relative position (OUTPOS) accordingly.

OVLY_OUT
(Can be used in FASTREXX procedures.) Overlays the output record with a literal (constant) or variable value.

OVLY_VAR
(Can be used in FASTREXX procedures.) Overlays the named character variable with a string.

PRINT
Prints a record.

PRTCOUNT
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Returns the count of records printed.

RECSIN
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Returns the count of records read.

RECSOUT
(Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Returns the count of records written to a given data set.

RSTR_OUT
(Can be used in FASTREXX condition expressions.) Restores the most recently saved copy of the output buffer.

SAVE_OUT
(Can be used in FASTREXX condition expressions.) Saves a copy of the current output buffer.

SET_OLEN
(Can be used in FASTREXX procedures.) Sets the length of the output record.

SETC (Can be used in FASTREXX procedures.) Defines or changes a character variable.

SETN (Can be used in FASTREXX procedures.) Defines or changes a numeric variable

TESTC
(Can be used in FASTREXX procedures.) Performs a conditional test against a character variable.

TESTN
(Can be used in FASTREXX procedures.) Performs a conditional test against a numeric variable.

TALLY
(Can be used in FASTREXX procedures.) Totals a field value and reports the total.

TFLD (Can be used in FASTREXX condition expressions or internally processed criteria expressions.) Searches a field in the input record for one or more occurrences of a string, or tests a field in the input record for one more numeric values. For dimensioned fields, you can search any or all of the elements of the array.

TM Tests a string for a bit value.

VAR_OUT
(Can be used in FASTREXX procedures.) Overlays the output record with a field from a variable.

VAR_TM
(Can be used in FASTREXX procedures.) Tests selected bits of a field in a variable.

WRITE
(Can be used in FASTREXX procedures.) Writes a record.

The following REXX external functions can only be used with DSEB (Data Set Edit Batch):

BOT (Can be used in FASTREXX procedures.) Move to the last record

DOWN
(Can be used in FASTREXX procedures.) Move down (forwards) a specified number of records

FINDNEXT
(Can be used in FASTREXX procedures.) Search for a string from the current record forwards

FINDPREV
(Can be used in FASTREXX procedures.) Search for a string from the current record backwards

RECCUR
(Can be used in FASTREXX procedures.) Return the current record number

TOP (Can be used in FASTREXX procedures.) Move to the first record

UP (Can be used in FASTREXX procedures.) Move up (backwards) a specified number of records

UPDATE
(Can be used in FASTREXX procedures.) Replace the current input record with the value in OUTREC

The following functions are only valid when used in template criteria for segmented templates.

SEGCNT
(Does not require REXX processing.) Returns the current segment number when identifying segments for a given record.

SEGNO
(Does not require REXX processing.) Returns the previous number of occurrences of a specified segment on the current record.

Note: You can only use these File Manager-specific REXX external functions, and the INREC and OUTREC variables, in a REXX procedure specified by the PROC parameter of a File Manager function (or, when using panels, by the **Use REXX proc** field). You cannot use these functions and variables in REXX procedures outside of this File Manager environment.

Absolute and relative positioning in external REXX functions

All File Manager external REXX functions that refer to positions within the input or output record can use absolute values to determine the byte location within the record. For example, the FLD function syntax is:

```
FLD(start_column,length,type)
```

where *start_column* can be an integer that refers to a specific byte in the input record.

However, some external REXX functions also allow positions to be specified as an offset value, relative to the “current position” within the input or output record. The current position is initialized as each record is processed but can be modified by these functions.

For input records, this allows you to perform tasks such as searching for a string in the input record and then testing or copying the contents of a field relative to the located string.

For output records, this allows you to easily append fields or constants at positions relative to the most recently updated output record field. For example, you could append a number of constants and fields, one after the other in the output record, without needing to keep track of the exact current starting position or manually updating the starting position with the length of added fields.

Note: You cannot use the REXX external functions to write data beyond the logical record limitations of the output data set. For example, when using a Fixed Block data set that has an LRECL of 80, you cannot write data to position 81 or beyond.

The following functions support relative positioning:
- CHG_OUT
- CHG_VAR
- FINDNEXT
- FINDPREV
- FLD
- FLD_CO
- FLD_OUT
- FLD_TM
- FLD_TYPE
- FLDI
- FLDO
- MOD_DATE
- OVLY_OUT
- OVLY_VAR
- SETC
- SETN
- TESTC
- TESTN
- VAR_OUT
- VAR_TM

To maintain the current position within the input and output records, these functions use two internal variables, INPOS and OUTPOS. These values have not been externalized to the REXX environment and can only be accessed or altered indirectly by using the functions listed above or SET_OLEN (which does not support relative position arguments but does alter the value of OUTPOS in some circumstances).

INPOS

For each new record that is processed, INPOS is set to 1. INTPOS is then modified (as a side effect) when the following functions are used:

FLD_CO

Whenever the FLD_CO function executes a successful search for a *needle* that is a string (type C or U), INPOS is then changed for the current input record. If a *needle* is found, INPOS is set to the first byte of the located *needle*. If no *needle* is found, INPOS is unchanged.

FLDI Whenever the FLDI function (with contains operator) executes a successful search for a *needle* that is a string (type C or U), INPOS is then changed for the current input record. If a *needle* is found, INPOS is set to the first byte of the located *needle*. If no *needle* is found, INPOS is unchanged.

When you use the FLDI function with the VER operator and the verification fails, INPOS is set to the column of the character that failed the verification.

FINDNEXT
Whenever the FINDNEXT function executes a successful search for a needle, the INPOS is changed and set to the first byte of the located needle. If no needle is found, then INPOS is unchanged.

FINDPREV
Whenever the FINDPREV function executes a successful search for a needle, the INPOS is changed and set to the first byte of the located needle. If no needle is found, then INPOS is unchanged.

OUTPOS
For each new record that is processed, OUTPOS is set to 1 greater than the length of the current output record. The output record is initially the same length as the input record, unless templates have been used to reformat the record. OUTPOS is then modified (as a side effect) when the following functions are used:

CHG_OUT
CHG_OUT sets OUTPOS to one byte past the end of the last changed field in the output record.

FLD_OUT
FLD_OUT sets OUTPOS to one byte past the end of the field being overlaid in the output record.

OVLY_OUT
OVLY_OUT sets OUTPOS to one byte past the end of the field being overlaid in the output record.

SET_OLEN
SET_OLEN only changes OUTPOS if it truncates the output record so that the existing OUTPOS becomes greater than the reduced record length. In this case, OUTPOS is reset to the reduced length plus 1.

FLDO with contains operator
Whenever the FLDO function (with contains operator) executes a successful search for a *needle* that is a string (type C or U), OUTPOS is then changed for the current input record. If a *needle* is found, OUTPOS is set to the first byte of the located *needle*. If no *needle* is found, OUTPOS is unchanged.

When you use the FLDO function with the VER operator and the verification fails, OUTPOS is set to the column of the character that failed the verification.

VARPOS
For each new record that is processed, VARPOS is set to 1. VARPOS is then modified (as a side effect) when the following functions are used:

CHG_VAR
CHG_VAR sets VARPOS to one byte past the end of the last changed field in the variable.

OVLY_VAR
OVLY_VAR sets VARPOS to one byte past the end of the field being overlaid in the variable.

TESTC

Whenever the TESTC function (with contains operator) executes a successful search for a *needle* that is a string (type C or U), VARPOS is then changed for the current variable. If a *needle* is found, VARPOS is set to the first byte of the located *needle*. If no *needle* is found, VARPOS is unchanged.

When you use the TESTC function with the VER operator and the verification fails, VARPOS is set to the column of the character that failed the verification.

Specifying relative positions

Relative positioning is specified when you use a special character string, in the form of <type><offset>, in place of an integer in the *start* argument of a supported function. The relative position can be specified in the following ways:

IP*x* The start position is taken from the current INPOS (the "I" in <type>) and then offset a positive (the "P" in <type>) number of bytes, as specified by *x*. For example, if the INPOS was currently 20 and you specified a *start* argument of IP5, the start position would be 25.

IN*x* The start position is taken from the current INPOS and then offset a negative (the "N" in <type>) number of bytes, as specified by *x*. For example, if the INPOS was currently 20 and you specified a *start* argument of IN5, the start position would be 15.

OP*x* The start position is taken from the current OUTPOS (the "O" in <type>) and then offset a positive number of bytes, as specified by *x*. For example, if the OUTPOS was currently 20 and you specified a *start* argument of OP5, the start position would be 25.

ON*x* The start position is taken from the current OUTPOS and then offset a negative number of bytes, as specified by *x*. For example, if the OUTPOS was currently 20 and you specified a *start* argument of ON5, the start position would be 15.

When the start argument's natural target is the input record, **IP***x* and **IN***x* can be abbreviated to **P***x* **or N***x*.

When the argument's natural target is the output record, **OP***x* and **ON***x* can be abbreviated to **P***x* **or N***x*.

When the argument's natural target is a variable, **IP***x*, **IN***x*, **OP***x*, **ON***x*, can be used to denote the current relative variable position. They can be abbreviated to **P***x* or **N***x*.

For example, `FLD(start_column,length,type)` reads from the input record, so you could specify *start_column* as IP5 or P5 and get the same result. On the other hand, if you wanted to use the current value of OUTPOS to specify the *start_column* in the input record, you must specify the <type> in full, that is, as OP5.

Using FASTREXX variables

File Manager supports these variables for FASTREXX processing:

- System numeric variables. See Table 19 on page 1101 below for descriptions.
- System character variables. See Table 18 on page 1101 below for descriptions.
- User character variables.

- User numeric variables.
- Tally registers.

System and tally variables are maintained by File Manager and are read-only to user procedures.

User variables are defined by SETC or SETN functions (there is an implied definition in TESTC and TESTN functions for variables that do not exist). These variables can be referenced and modified by these functions:

Table 17. Batch update status and action

Function	Reference	Modify
CHG_VAR	Y	Y
CHG_OUT	Y	N
FLDI	Y	N
FLDO	Y	N
OVLY_VAR	Y	Y
OVLY_OUT	Y	N
SETC	Y	Y
SETN	Y	Y
TESTC	Y	N
TESTN	Y	N
VAR_OUT	Y	N
VAR_TM	Y	N

A user variable persists from its creation by a procedure to the end of the File Manager invocation. This allows any number of procedures run within the same File Manager invocation to refer to the same variables.

Table 18. System character variables

Name	Description
ZINREC	Input record
ZOUTREC	Output record
ZMEMBERI	Input member name
ZMEMBERO	Output member name
ZDSNIN	Input data set name
ZDSNOUT	Output data set name

Table 19. System numeric variables

Name	Description
ZRECSIN	Input record count
ZRECSOUT	Output record count

Tally register for external REXX functions

Table 20 on page 1102 shows the functions that support a tally register that allows you to report on the function activity.

Table 20. Functions supporting a TALLY register

Function name	Counts number of	Sample coding for tally literal
CHANGE	Strings changed	`(fld(1),'a','c',0,,,'Change 'a' to 'c' ')`
CHG_OUT	Strings changed	`chg_out('a','c',0,,,, 'Change 'a' to 'c' ')`
CHG_VAR	Strings changed	`chg_var(myvar,'a','c',0,,,, 'Change 'a' to 'c' ')`
CONTAINS	True results	`co(fld(1,2),'aa','bb','cc',,'Contains 'aa','bb','cc'')`
FLD_CO	True results	`fld_co(1,2,c,'aa','bb','cc',, 'Contains 'aa','bb','cc'')`
FLD_OUT	Invocations	`fld_out(1,2,3,2,,'Move Columns 1,2 to Columns 3,4 ')`
FLD_TM	True results	`fld_tm(1,'01'x,,'Test under mask column 1 for '01'x ')`
FLD_TYPE	True results	`fld_type(36,1,Z,'Check Column 36 for valid zoned')`
FLDI	True results	`fldi(1,4,b,'>',64,'People over 64')`
FLDO	True results	`fldo(1,4,b,'>',64,'People over 64')`
NCONTAIN	True results	`nco(fld(36,1),1,4,3,2,,Column 36 contains 1,4,3,2'))`
OFLD_CO	True results	`ofld_co(1,2,c,'aa','bb','cc',, 'Output contains 'aa','bb','cc'')`
OVLY_OUT	Invocations	`ovly_out('**',1,2,,,'Overlay columns 1,2 with '**'  ')`
OVLY_VAR	Invocations	`ovly_var(myvar,'**',1,2,,,'Overlay columns 1,2 with '**'  ')`
SET_OLEN	Invocations	`Set_olen(84,'b','Change output record length to 84')`
SETC	True results	`setc(myvar,'abc',,'Set myvar to abc')`
SETN	True results	`setn(mynum,'+2','Add 2 to mynum')`
TESTC	True results	`testc(myname,'cu','Smith','Jones',,'Common surnames')`
TESTN	True results	`testn(varage,'>',64,'People over 64')`

TFLD	True results	`tfld('Age','>',64,'People over 64')`
		`tfld('Age','NN','Non-Numeric Age fields')`
		`tfld('Age','RG',21,75,'People between 21 and 75' )`
		`tfld('Name','CU','Smith','Jones',,'Common surnames')`

TM	True results	`tm(fld(1,1),'01'x,,'Test under mask column 1 for '01'x ')`
VAR_OUT	Invocations	`var_out(myvar,1,2,3,2,,'Move Columns 1,2 to Columns 3,4  ')`
VAR_TM	True results	`var_tm(myvar,1,'01'x,,'Test under mask column 1 for '01'x  ')`

Specifying your tally register

The tally register is defined when you provide a literal value as an additional operand to the functions in Table 20. For functions that have a fixed number of operands, the literal operand is the next positional operand beyond the defined operands for the given function. For functions that have an indefinite number of operands, a null operand is required to delimit the function operands and to denote the next operand as a tally literal.

Example 1

```
IF FLD_CO(1,8,c,'a',,'Number of records with "a"') then
  chg_out('a','c',0,,,,'Number of strings changed from "a" to "c"')
```

produces this tally report:

```
TALLY summary report
-----------------------------------------------------------
Number of records with "a"                       4
Number of strings changed from "a" to "c"     32
```

Note: FLD_CO can have an indefinite number of search literals. As a result, the tally register is specified by `,,'Number of records with "a"`.

For CHG_OUT, no null positional delimiter is required and the TALLY literal must be the seventh operand.

Example 2

```
*FASTREXX
if fld_tm(1,'01'x) then do;
  OVLY_OUT('**',1,2,,, 'Count of first 2 chars set to "**"')
  return
end;
```

produces this tally report:

```
TALLY summary report
-----------------------------------------------------
Count of first 2 chars set to "**"     4
```

Using a tally register

Each tally register is defined by the literal description. If you code the same literal description in a number of functions, then the same tally registered is incremented as determined by the function being invoked.

BOT (DSEB only)

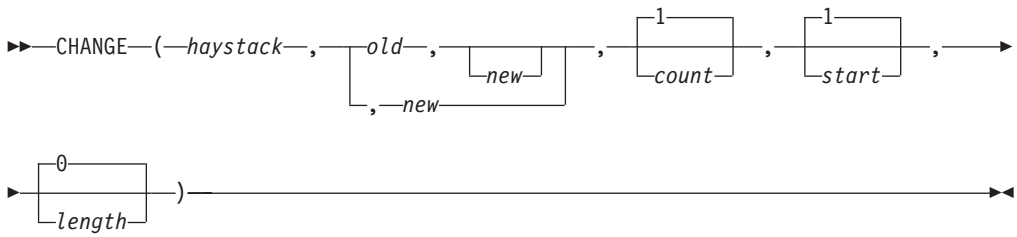
Syntax

(Can be used in FASTREXX procedures.)

Moves to the last input record.

CHANGE

Syntax



Note: Commas following the last specified argument can be omitted.

Searches *haystack* and changes one or more occurrences of *old* to *new*.

Returns

Returns *haystack*, with up to *count* occurrences of the substring *old* changed to the substring *new*.

haystack

The string that you want to search.

old Old string to change. If this argument is omitted, the new string is inserted at the *start* location.

new New string. If this argument is omitted, then *count* occurrences of *old* are deleted.

count Maximum number of occurrences of *old* to change. Must be a non-negative integer. Default value is 1. A value of 0 indicates that all occurrences should be changed, unless the *old* string field is omitted, in which case it is equivalent to a value of 1.

start Position in *haystack* in bytes at which to start searching for occurrences of *old*. Must be a positive integer. The default value is 1. If *start* is greater than the current length of the output record, the function has no effect.

length Number of bytes within *haystack* to search for occurrences of *old*. Must be a non-negative integer. A value of 0 indicates that the remainder of *haystack* from *start* should be searched. If *length* is less than the length of *old*, the function has no effect.

Example 1

```
CHANGE('abcabcabc','abc','DeF')      ►    'DeFabcabc'
       /*  1 (default) occurrence of old changed          */
```

Example 2

```
CH('abcabcabc','abc','DeF',2)        ►    'DeFDeFabc'
       /*  2 occurrences of old changed                   */
```

Example 3

```
CHANGE('abcabcabc','abc','DeF',0)    ►    'DeFDeFDeF'
      /*  count = 0, all occurrences of old changed       */
```

Example 4

```
CH('abcabcabc','abc','DeF',,4)       ►    'abcDeFabc'
      /*  1 (default) occurrences of old changed,         */
      /*  starting at position 4                          */
```

Example 5

```
CHANGE('aaaaaaaa','a','A',0,3,2)     ►    'aaAAaaaa'
      /*  all occurrences of old changed, starting at     */
      /*  position 3 for a length of 2                    */
```

Example 6

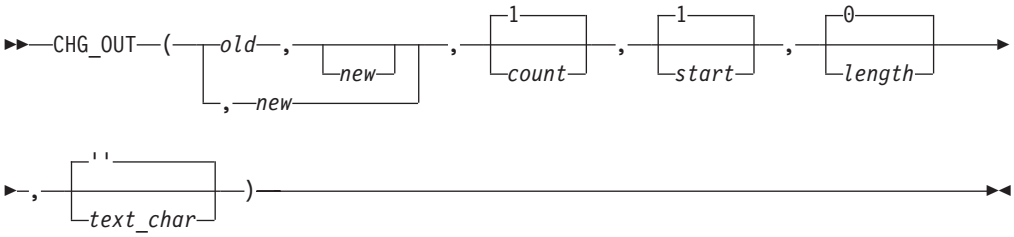
```
CH('abcabcabc','a',,0)               ►    'bcbcbc'
      /*  new omitted, count = 0,                         */
      /*  all occurrences of old deleted                  */
```

Example 7

```
CHANGE('abc',,'def',,2)              ►    'adefbc'
      /*  old omitted, new inserted, starting at          */
      /*  position 2                                      */
```

CHG_OUT

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX procedures.

Changes one or more occurrences of an *old* string in the output record to a *new* string. On successful execution, also updates the value of OUTPOS to one byte past the end of the last changed field in the output record.

Returns
A single blank.

old Old string to change. If this argument is omitted, the new string is inserted at the *start* location.

You can substitute a character or numeric variable or tally literal by specifying an *&varname* where *varname* matches an existing variable name.

Note:

1. Numeric values are converted to display form with leading zeros removed.
2. If a variable name is not found, then the string is interpreted as literal.

new New string. If this argument is omitted, then *count* occurrences of *old* are deleted.

You can substitute a character or numeric variable or tally literal by specifying an *&varname* where *varname* matches an existing variable name.

Note:

1. Numeric values are converted to display form with leading zeros removed.
2. If a variable name is not found, then the string is interpreted as literal.

count Maximum number of occurrences of *old* to change. Must be a non-negative integer. Default value is 1. A value of 0 indicates that all occurrences should be changed, unless the *old* string field is omitted, in which case it is equivalent to a value of 1.

start Position, in bytes, in the output record at which to start searching for occurrences of *old*. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the output record, the function has no effect.

Relative to current INPOS
Must be specified as IP*x* or IN*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the output record, the function has no effect.

Relative to current OUTPOS
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*.If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the output record, the function has no effect.

length Amount, in bytes, of the output record to search for occurrences of *old*. Must be a non-negative integer. Default value is 0. A value of 0 indicates that the remainder of the output from *start* should be searched. If *length* is less than the length of *old*, the function has no effect.

text_char
Can be a null string or a single character.

Specifying a null string (the default), indicates CHG_OUT should behave without text sensitivity.

Specifying a single character defines the special text-sensitive character and indicates that text-sensitive change behaviour is required. When a character has been specified for *text_char*, CHG_OUT behaves as follows:

- If the new and old strings are the same length, CHG_OUT behaves as if text-sensitive change behaviour had not been requested.
- If the new string is shorter than the old string, then when a replacement is made, CHG_OUT searches for the first *text_char* character following the end of the replaced string. Note that the entire record is searched. If the *text_char* character is found, additional *text_char* characters are inserted at the point of this first subsequent character, to make up the difference in length between new and old. If the *text_char* character is not found in the remainder of the record, then no insertion takes place and the record is reduced in length. If the record is fixed-length and no subsequent action occurs to make up the shortfall, then the File Manager record padding process fills out the record when it is written.

  The intended effect is that text on multiple lines, if aligned in columns separated by the text char character, continues to align in columns after replacement. This is useful for updating files with sequence numbers on the right, such as COBOL or JCL.
- When the new string is longer than the old string., then when a replacement is made, CHG_OUT searches to the right of the replaced string for two consecutive *text_char* characters. Note that the entire record is searched. If two consecutive *text_char* characters are found, they are replaced with a single *text_char* character. This process is repeated, starting from the remaining single character (and including that character) until the length difference between old and new is accounted for. Using this algorithm, multiple *text_char* characters can be reduced to a single *text_char* character, but a single *text_char* char character between other characters is never eliminated.

  The intended effect is to try to use existing “blank” areas in the string, to leave text on the right unchanged as much as possible. There is no guarantee that there will be an adequate number of *text_char* characters to accomplish the goal. If there are not enough *text_char* characters, then the rest of the record is shifted right and possibly truncated when it is written, if fixed in length. This is useful for modifying files like COBOL or JCL source, where a sequence number may exist to the right.
- If you have specified multiple string replacements (*count* is greater than 1), then the intent remains the same. The search proceeds from left to right, firstly checking for the search argument, and secondly *text_char*

characters to expand or collapse. If the search argument is found, it is replaced and the search continues immediately following the replaced string.

- Note that the string replacement may be limited to byte positions by the *length* argument. However, the search for *text char* characters to add or remove continues past that limit to the end of the record if required.

Example 1

Assuming that the current output record contains 'abcabcabcabcabcabcabc', then:

```
CHG_OUT('abc','DeF',0)
/*  All occurrences of old within the           */
/*  output record are changed                   */
```

The output record becomes 'DeFDeFDeFDeFDeFDeFDeF'.

Example 2

Assuming that the current output record contains 'abcabcabcabcabcabcabc', then:

```
CHG_OUT('abc','DeF',,4)
/*  1 (default) occurrences of old changed,     */
/*  starting at position 4 within the output record      */
```

The output record becomes 'abcDeFabcabcabcabcabc'.

Example 3

Assuming that the current output record contains 'aaaaaaaaaa', then:

```
CHG_OUT('a','A',0,3,2)
/*  all occurrences of old changed, starting at  */
/*  position 3 in the output record, for a length of 2    */
```

The output record becomes 'aaAAaaaaaa'.

Example 4

Assuming that the current output record contains 'abcabcabcabcabcabcabc', and that INPOS is currently set to 13 and OUTPOS is currently set to 4, then:

```
CHG_OUT('abc','DeF',1,P3)
/*  1 occurrence of old changed,                */
/*  uses OUTPOS as the default target, therefore         */
/*  starts at position 7 within the output record        */
```

The output record becomes 'abcabcDeFabcabcabcabc' and OUTPOS is set to 10 (INPOS remains unchanged).

Example 5

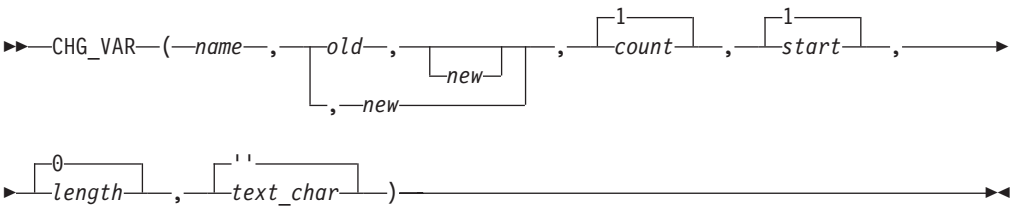
Assuming that the current output record contains 'abcabcabcabcabcabcabc', and that INPOS is currently set to 13 and OUTPOS is currently set to 4, then:

```
CHG_OUT('abc','DeF',1,IN3)
/*  1 occurrence of old changed,                */
/*  forces start to use INPOS value, therefore           */
/*  starts at position 10 within the output record       */
```

The output record becomes 'abcabcabcDeFabcabcabc' and OUTPOS is set to 13 (INPOS remains unchanged).

CHG_VAR

Syntax



(Can be used in FASTREXX procedures.)

Note: Commas following the last specified argument can be omitted.

Changes one or more occurrences of an old string in the variable to a new string. On successful execution, also updates the value of OUTPOS to one byte past the end of the last changed field in the variable.

Returns
A single blank.

name 1–256 character variable identifier. Variable name matching is not case sensitive. If the name is not found, a severe error occurs and the procedure is terminated. Cannot be a system character variable or a system numeric variable. See “Using FASTREXX variables” on page 1100.

old Old string to change. If this argument is omitted, the new string is inserted at the start location. You can substitute a character or numeric variable or tally literal by specifying an *&varname* where *varname* matches an existing variable name.

Note:

1. Numeric values are converted to display form with leading zeros removed.
2. If a variable name is not found, then the string is interpreted as a literal.

new New string. If this argument is omitted, then *count* occurrences of *old* are deleted. You can substitute a character or numeric variable or tally literal by specifying an *&varname* where *varname* matches an existing variable name.

Note:

1. Numeric values are converted to display form with leading zeros removed.
2. If a variable name is not found, then the string is interpreted as a literal.

count Maximum number of occurrences of *old* to change. Must be a non-negative integer. Default value is 1. A value of 0 indicates that all occurrences should be changed.

start Position, in bytes, in the variable at which to start searching for occurrences of *old*. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the variable, the function has no effect.

Relative to current variable position
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*, or as IP*x* or IN*x*. If this resolves to a value of less than or equal to zero, the function results in an error.If this resolves to a value that is greater than the current length of the variable, the function has no effect.

length Amount, in bytes, of the variable to search for occurrences of *old*. Must be a non-negative integer. Default value is 0. A value of 0 indicates that the remainder of the output from *start* should be searched. If *length* is less than the length of *old*, the function has no effect.

text_char
Can be a null string or a single character.

Specifying a null string (the default), indicates CHG_VAR should behave without text sensitivity.

Specifying a single character defines the special text- sensitive character and indicates that text-sensitive change behaviour is required. When a character has been specified for text_char, CHG_VAR behaves in this way:

- If the new and old strings are the same length, CHG_VAR behaves as if text-sensitive change behaviour had not been requested.
- If the new string is shorter than the old string, then when a replacement is made, CHG_VAR searches for the first *text_char* character following the end of the replaced string. Note that the entire record is searched. If the *text_char* character is found, additional *text_char* characters are inserted at the point of this first subsequent character, to make up the difference in length between *new* and *old*. If the *text_char* character is not found in the remainder of the record, then no insertion takes place and the record is reduced in length. If the record is fixed-length and no subsequent action occurs to make up the shortfall, then the File Manager record padding process fills out the record when it is written.

  The intended effect is that text on multiple lines, if aligned in columns separated by the text char character, continues to align in columns after replacement. This is useful for updating files with sequence numbers on the right, such as COBOL or JCL.
- When the new string is longer than the old string, then when a replacement is made, CHG_VAR searches to the right of the replaced string for two consecutive *text_char* characters. Note that the entire record is searched. If two consecutive *text_char* characters are found, they are replaced with a single *text_char* character. This process is repeated, starting from the remaining single character (and including that character) until the length difference between *old* and *new* is accounted for. Using this algorithm, multiple *text_char* characters can be reduced to a single *text_char* character, but a single *text_char* character between other characters is never eliminated.

  The intended effect is to try to use existing "blank" areas in the string, to leave text on the right unchanged as much as possible. There is no guarantee that there will be an adequate number of *text_char* characters to accomplish the goal. If there are not enough *text_char* characters, then the rest of the record is shifted right and possibly truncated when it is written, if fixed in length. This is useful for modifying files like COBOL or JCL source, where a sequence number may exist to the right.

- If you have specified multiple string replacements (*count* is greater than 1), then the intent remains the same. The search proceeds from left to right, firstly checking for the search argument, and secondly *text_char* characters to expand or collapse. If the search argument is found, it is replaced and the search continues immediately following the replaced string.
- Note that the string replacement may be limited to byte positions by the length argument. However, the search for *text_char* characters to add or remove continues past that limit to the end of the record if required.

Example 1

Assuming that the current variable contains 'abcabcabcabcabcabcabc', then:

```
CHG_VAR(MYVAR,'abc','DeF',0)
/*  All occurrences of old within the            */
/*  variable are changed                         */
```

The variable becomes 'DeFDeFDeFDeFDeFDeFDeF'.

Example 2

Assuming that the current variable contains 'abcabcabcabcabcabcabc', then:

```
CHG_VAR(MYVAR,'abc','DeF',,4)
/*  1 (default) occurrences of old changed,      */
/*  starting at position 4 within the variable   */
```

The variable becomes 'abcDeFabcabcabcabcabc'.

Example 3

Assuming that the current variable contains 'aaaaaaaaaa', then:

```
CHG_VAR(MYVAR,'a','A',0,3,2)
/*  all occurrences of old changed, starting at  */
/*  position 3 in the variable, for a length of 2 */
```

The variable becomes 'aaAAaaaaaa'.

Example 4

Assuming that the current variable contains 'abcabcabcabcabcabcabc', and that current variable position is 4, then:

```
CHG_VAR(MYVAR,'abc','DeF',1,P3)
/*  1 occurrence of old changed,                 */
/*  uses current position as the default target, therefore  */
/*  starts at position 7 within the variable     */
```

The variable becomes 'abcabcDeFabcabcabcabc' and the variable position is set to 10.

Example 4

Assuming that the current variable contains 'abcabcabcabcabcabcabc', and that current variable position is currently set to 13 , then:

```
CHG_VAR(MYVAR,'abc','DeF',1,IN3)
/*  1 occurrence of old changed,                 */
/*  forces start to use current position value, therefore   */
/*  starts at position 10 within the variable    */
```

The variable becomes 'abcabcabcDeFabcabcabc' and the variable position is set to 13.

CONTAINS

Syntax

Checks the contents of *haystack* for one or more occurrences of *needle*.

Returns

If the *haystack* string contains one or more of the *needle* strings, then CONTAINS returns 1. Otherwise, CONTAINS returns 0.

CONTAINS is case-sensitive: it only returns 1 if the *haystack* contains a string with the same mix of uppercase and lowercase as a *needle*.

haystack
The string that you want to search.

needle The string that you are attempting to find within *haystack*. You can search for up to 20 strings at a time.

For a similar function that matches numeric values, see “NCONTAIN” on page 1128. For a FASTREXX-eligible equivalent, see “FLD_CO” on page 1114.

Example 1

If the current input record contains “Michael”, “Mick” or “Mike” in the first ten columns, then print the record.

```
If CO(FLD(1,10),'Michael','Mick','Mike') Then
  PRINT(inrec, 'CHAR')
```

Example 2

If the current input record contains “USA”, “Australia” or “England”, then drop the record from processing.

```
If CONTAINS(inrec,'USA','Australia','England') Then
  Return 'DROP'
```

DOWN (DSEB only)

Syntax

(Can be used in FASTREXX procedures.)

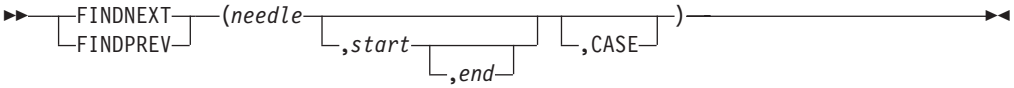
Moves down *n* number of records, or to the last record, if there are less than *n* records below the current record.

Returns

If, before moving, the current record is the last record, then the DOWN function returns the string value “EOF” (end of file). Otherwise, DOWN returns 0.

FINDNEXT, FINDPREV (DSEB only)

Syntax



(Can be used in FASTREXX procedures.)

Searches for *needle* in the input data set, from the current input record forwards (FINDNEXT) or backwards (FINDPREV). You can limit the search to a range of columns, or to an exact matching case.

If a FINDNEXT() or FINDPREV() has located a needle on a given record, then repeated searching resumes from the previously located needle plus 1 for FINDNEXT and minus 1 for FINDPREV. If the record location has been changed by an UP(), Down(), TOP(), or BOT() command, or there has been no previous located needle, then FINDPREV searches from the end of the current record backward, and FINDNEXT searches from the start of the current record forward.

Returns

If the search is successful, the record in which *needle* was found becomes the current input record, and the FINDNEXT function returns the starting column of *needle* in the record. If the search is unsuccessful, the current input record remains the same, and FINDNEXT returns 0.

needle String(s) or numeric(s) to search for. The needle can be a character variable in the form &*varname* where *varname* matches an existing variable name. If an "&" is coded and a variable name is not found, then the needle is treated as the string value to be searched for.

start The position, in bytes, of the start of the range in each input record to be searched.

end The position, in bytes, of the end of the range in each input record to be searched.

CASE Specifies that the comparison is case-sensitive.

If you want to save any changes you have made to the record that was current prior to calling FINDNEXT or FINDPREV, use the UPDATE function. Otherwise, if the search is successful, any changes made to that record are lost when FINDNEXT or FINDPREV moves to another record.

Here are some examples:

Example 1

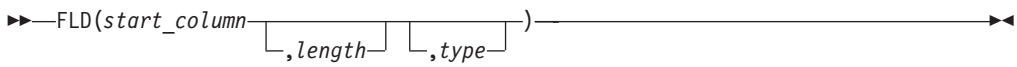
```
FINDPREV('abc') /* Finds 'abc', 'ABC', 'ABc' and so on */
```

Example 2

```
FINDNEXT('abc',1,10,'CASE') /* Finds 'abc', but not 'ABC', 'ABc' and so on */
```

FLD

Syntax



Can be used in FASTREXX condition expressions.

Fetches the value of a field from the current input record (INREC), starting at *start_column*, of *length* number of bytes, interpreted according to the specified *type*.

Returns

The value of the field from the current input record.

start_column

Position, in bytes, in the input record at which to start reading the field value. Can be specified as:

Absolute position

Must be a positive integer. Default value is 1. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS

Can be specified as IP*x* or IN*x*, or as P*x* or N*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS

Must be specified as OP*x* or ON*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

length The length of the field in bytes.

For binary fields, you must specify the length. It can be 2, 4, or 8.

For character fields, if you omit the length, FLD returns the remainder of the record.

For packed decimal fields, if you specify the length, it must be in the range 1–16. If you omit the length, FLD attempts to determine the packed field length from the record data and returns only that field.

For zoned decimal fields, if you specify the length, it must be in the range 1–31 or, if the field contains a separate sign character, in the range 1–32. If you omit the length, FLD returns the remainder of the record.

type The data type of the field. Valid values are:

B Binary. FLD interprets binary fields as being signed.

C Character. This is the default.

P Packed decimal.

U Interprets the field as Character, but converts it to uppercase before returning the string.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

If you specify a value for *length* that would cause the record length to be exceeded:

- For character fields (*type* C, U), FLD returns the remainder of the record.
- For numeric fields (*types* B, P, Z), FLD returns a null string.

If you specify a numeric type (*types* B, P, Z), and the specified field contains invalid data for that type, then FLD returns a null string. Numeric data is always returned in integer form; that is, FLD does not perform scaling of numeric data.

The FLD function is similar to the built-in REXX SUBSTR function, except that FLD interprets the "substring" according to the specified data type, and returns the value formatted appropriately. (For a numeric field, FLD returns the value with a sign, and without leading zeros.)

Example 1

If the value of the packed decimal field that starts at column 8 is greater than 100, then do not process the current record.

```
If FLD(8,P) > 100 Then Return 'DROP'
```

Example 2

If the value of the 2-digit year field starting at column 42 is greater than 60, then insert the literal "19" before the year field; otherwise, insert "20".

```
If FLD(42,2,Z) > 60 Then
   outrec = FLD(1,41)||'19'||FLD(42)
Else
   outrec = FLD(1,41)||'20'||FLD(42)
```

Example 3

If the 4-byte field that starts at column 11 does not contain valid packed decimal data, then do not process the current record.

```
If FLD(11,4,p) = '' Then Return 'DROP'
```

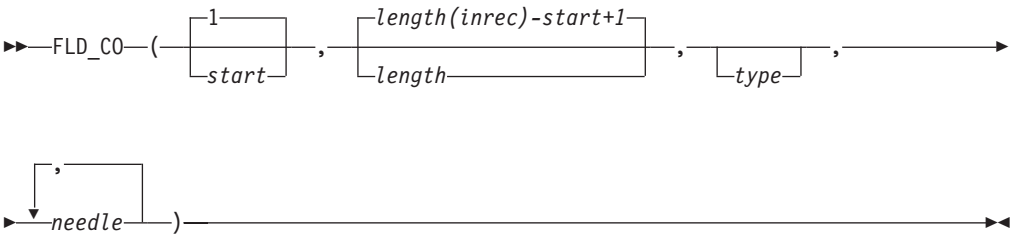
Example 4

If the value of the packed decimal field that starts at INPOS + 8 is greater than 100, then do not process the current record.

```
If FLD(P8,P) > 100 Then Return 'DROP'
```

FLD_CO

Syntax



(Can be used in FASTREXX condition expressions.)

Searches the field within the input record specified by *start* and *length*, for one or more occurrences of *needle*. On successful execution when searching for a string, also updates the value of INPOS to the first byte of the located field in the input record.

Returns
If at least one occurrence of *needle* is found, returns 1. If no occurrences are found, returns 0.

start Position, in bytes, in the input record at which to start searching for occurrences of *needle*. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS
Must be specified as OP*x* or ON*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

length Length, in bytes, of the field to be searched.

- For character fields, the length defaults to the remaining record length from the start position to the end of the record (inclusive). A value of 0 also indicates that the field extends to the end of the record.
- For binary fields, you must specify the length. It can be 2, 4, or 8.
- For packed decimal fields, if you specify the length, it must be in the range 1-16. If you omit the length, FLD_CO attempts to determine the packed field length from the record data.
- For zoned decimal fields, if you specify the length, it must be in the range 1-31 or, if the field contains a separate sign character, in the range 1-32. If you omit the length, it defaults to the remainder of the record.

type The data type of the field. Valid values are:

B Binary. FLD_CO interprets binary fields as being signed.

C Character. This is the default. The comparison is case sensitive.

P Packed decimal.

U Interprets the field as Character, but converts it to uppercase before comparing it with *needle*.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

needle String(s) or numeric(s) to search for. For the character types, FLD_CO searches the *haystack* for each of the *needles*. In this context, it behaves like a combination of the FLD and CONTAINS functions. For the numeric data types, the *haystack* is treated as a single numeric field, and an appropriate numeric comparison is performed against each of the *needles*. In this

context it behaves like a combination of the FLD and NCONTAIN functions. You can search for up to 20 strings at a time.

To perform case-insensitive searches, specify *type* as 'U' and *needle* in uppercase.

Example 1

If the current input record contains “MIKE”, “Mike” or “mike” in the first ten columns, then write the record.

```
If FLD_CO(1,10,'U','MIKE') Then WRITE('MDD')
```

Example 2

If the current input record contains “USA”, “Australia” or “England”, then drop the record from processing.

```
If FLD_CO(,,,'USA','Australia','England') Then Return 'DROP'
```

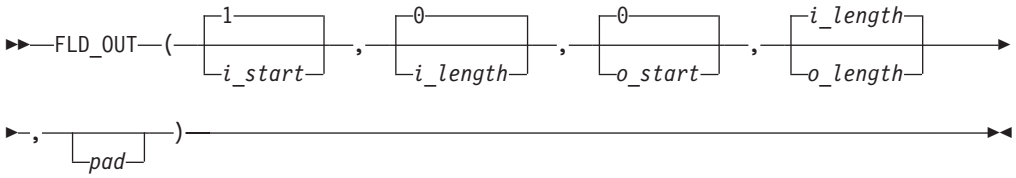
Example 3

If the current input record contains “MIKE”, “Mike” or “mike” in the ten columns immediately after the current INPOS, then write the record.

```
If FLD_CO(P0,10,'U','MIKE') Then WRITE('MDD')
```

FLD_OUT

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX procedures.

Overlays the output record with a field from the input record. See OVLY_OUT for a function to overlay the output record with a literal. If the target field length exceeds the source field length, then the source field is padded to the specified length using the pad character. If the target field length is less than the source field, the source field is truncated from the right. On successful execution, also updates the value of OUTPOS to one byte past the end of the field overlaid in the output record.

Returns
A single blank

i_start Position, in bytes, in the input record at which to start reading the field to be copied. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*. Must resolve to a positive integer.

Relative to current OUTPOS
Must be specified as OP*x* or ON*x*. Must resolve to a positive integer.

i_length
Length, in bytes, of the source field. Must be a non-negative integer. Defaults to 0. If you omit *i_length* or specify zero, the remainder of the input record from the *i_start* position is used. This also applies if you specify a value that would cause the source field to be read from beyond the end of the current input record.

o_start Position, in bytes, in the output record at which to start overlaying the copied field. If you omit *o_start* or specify zero, the field is appended to the end of the output record. If *o_start* is greater than the current length of the output record, the record is padded with the specified or defaulted pad character from the current record length to the specified start position. Can be specified as:

Absolute position
Must be a non-negative integer. Default value is 0.

Relative to current INPOS
Must be specified as IP*x* or IN*x*. Must resolve to a positive integer.

Relative to current OUTPOS
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*. Must resolve to a positive integer.

o_length
Length, in bytes, of the target field. Defaults to the source field length (*i_length*). A value of 0 indicates that the target field length is the greater of *i_length* and the remaining output record length. If 0 is specified for both *o_start* and *o_length*, then *i_length* is used as the target length.

pad Pad character. Defaults to the pad character set on the File Manager System Processing Options panel (when processing online) or the pad character specified in the SET function (when running in batch). If the current pad setting is OFF or unspecified, the default pad character is a blank.

Example 1

Copy the characters in columns one and two of the input record to columns three and four of the output record.

```
FLD_OUT(1,2,3,2)
```

Example 2

Append the characters in columns eleven and twelve of the input record to the end of the output record, padded with two blanks.

```
FLD_OUT(11,2,0,4,' ')
```

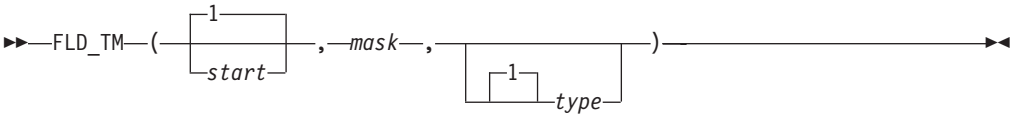
Example 3

Search a field in the input record for the characters 'AA' and, if found, copy to the end of the output record (assumes that OUTPOS is still set to end of output record).

```
IF FLD_CO(10,2,C,'AA') Then
/* when successful, updates INPOS to 12 */
 FLD_OUT(N2,2,P0,2)
/* copies from INPOS - 2, appends to end of output record */
```

FLD_TM

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX condition expressions.

Tests selected bits of a field in the input record.

Returns
Returns 1 if the test evaluates as True, and 0 if the test evaluates as False.

start Position, in bytes, in the input record at which to start testing. The length of the field is defined by the *mask*. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS
Must be specified as OP*x* or ON*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

mask Bit-string determining which bits to test in the field. The length of the mask defines the length of the input field. This field defines a bit-string mapping used to test the specified bits in the input record. You can use the bit-string, hex-string or character-string formats to define this field, Therefore, '0100 0000'b, '40'x, and ' ' are all legitimate and equivalent ways of defining a mask to test the second bit of a one-byte field.

type Type of test.

1 FLD_TM returns True (1) if **all** the bits that are **on** in the mask are **on** in the input record field. This is the default value.

0 FLD_TM returns True (1) if **all** the bits that are **on** in the mask are **off** in the input record field.

M FLD_TM returns True (1) if **at least one** of the bits that are **on** in the mask is **on** in the input record, and **at least one** is **off**.

N FLD_TM returns True (1) if **at least one of** the bits that are **on** in the mask is **off** in the input record field.

Example 1

Test the third byte of the input record and, if the low order bit is set, overlay a hex FF into the second byte of the output record.

```
If FLD_TM(3,'01'x) Then Do
  OVLY_OUT('ff'x,2,1)
  Return
End
Return 'DROP'
```

Example 2

Test the third byte of the input record and if some of the three high order bits are set, and some are not, overlay the contents of the second byte of that record with a hex 04.

```
If FLD_TM(3,'11100000'b,M) Then Do
  OVLY_OUT('04'x,2,1)
```

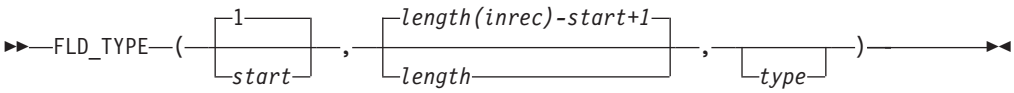
Example 3

Test the current INPOS position of the input record and, if the low order bit is set, overlay a hex FF into the byte prior to this location in the output record.

```
If FLD_TM(P0,'01'x) Then Do
  OVLY_OUT('ff'x,IN1,1)
  Return
End
Return 'DROP'
```

FLD_TYPE

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX condition expressions.

Tests the data type of a field in the input record.

Returns

Returns 1 if the test evaluates as True, and 0 if the test evaluates as False.

start Position, in bytes, in the input record at which to start testing. Can be specified as:

Absolute position

Must be a positive integer. Default value is 1. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS
Must be specified as OP*x* or ON*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

length Length of the field in the input record.

- For packed decimal fields, if you specify the length, it must be in the range 1-16. If you omit the length, FLD_CO attempts to determine the packed field length from the record data.
- For zoned decimal fields, if you specify the length, it must be in the range 1-31 or, if the field contains a separate sign character, in the range 1-32. If you omit the length, it defaults to the remainder of the record. A value of 0 also indicates that the field extends to the end of the record.

type Data type to test for.

P FLD_TYPE returns 1 if the field is a valid packed decimal field. Variant sign values (such as 'f'x for positive) are considered valid. Returns 0 otherwise.

Z FLD_TYPE returns 1 if the field is a valid zoned decimal field. FLD_TYPE recognizes all of the COBOL external decimal variants as numeric data. Returns 0 otherwise.

Example 1

If the first three bytes of the current input record contain a valid packed decimal number, tally the field. Otherwise, tally the first two bytes as a binary number.

```
If FLD_TYPE(1,3,P) Then
  TALLY(1,3,P,'Tally packed')
Else
  TALLY(1,2,B,'Tally binary')
```

Example 2

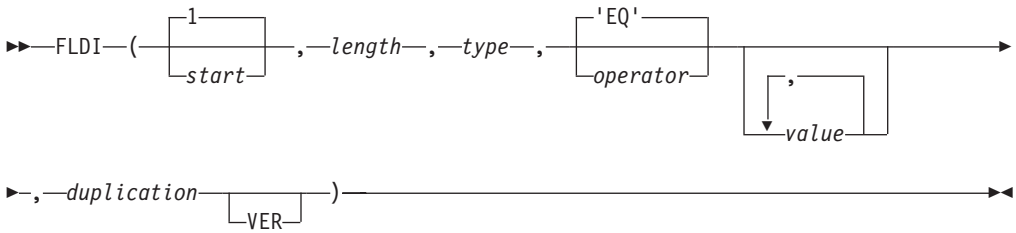
If the three bytes starting at the current INPOS in the input record contain a valid packed decimal number, copy the three bytes to the end of the output record. Otherwise, copy the two bytes starting at INPOS to the end of the output record.

```
If FLD_TYPE(P0,3,P) Then
  FLD_OUT(P0,3,P0,3)
Else
  FLD_OUT(P0,2,P0,2)
```

Note: In this example, the abbreviated form of the relative position specification can be used in both arguments of the FLD_OUT function. This is because *i_start* naturally targets the input record and *o_start* naturally targets the output record.

FLDI

Syntax



Can be used in FASTREXX condition expressions.

Performs a conditional test against input record field defined in by the start length and type parameters.

Note:

1. The operator (operator), and non-numeric values should all be enclosed in quotes to avoid syntax errors.
2. If you specify a value for *length* that would cause the record length to be exceeded, a false result is returned.
3. If you specify a numeric type (types B, P, Z), and the specified field contains invalid data for that type, then the function returns a false result. Numeric data is always returned in integer form; that is, the function does not perform scaling of numeric data.

start Position in bytes, in the input record at which to start reading the field value. Can be specified as:

Absolute position
Must be a positive integer. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS
Can be specified as IPx or INx, or as Px or Nx. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function produces a false result.

Relative to current OUTPOS
Must be specified as OPx or ONx. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function produces a false result.

length The length of the field in bytes.

- For binary fields, you must specify the length. It can be 2, 4, or 8.
- For character fields, if you omit the length, the length is defaulted:
  - For contains type operators, to the rest of the input record.
  - When all values are variable substitutions, to the rest of the input record.
  - Otherwise, the maximum literal value length is be used.

- For packed decimal fields, if you specify the length, it must be in the range 1–16. If you omit the length, the function attempts to determine the packed field length from the record data and returns only that field.
- For zoned decimal fields, if you specify the length, it must be in the range 1–31 or, if the field contains a separate sign character, in the range 1–32. If you omit the length, the function returns the remainder of the record. If this exceeds 32, then the function returns a false result.

type The data type of the field. Valid values are:

B Binary. The function interprets binary fields as being signed.

C Character. This is the default.

P Packed decimal.

U Interprets the field as character, but converts it to uppercase before returning the string.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

operator

The default is EQ or =. This function supports all the operators described for dynamic template and criteria edit. For details about the operators supported and their description, see:

- "Dynamic Template panel" on page 529.
- "Record Identification Criteria panel" on page 625.
- "Record Selection Criteria panel" on page 631.

value The value or values entered must be valid in the context of the operator and the field which is being referenced. For example, only certain operators like CO (contains) allow multiple values. Numeric values should be entered when testing numeric fields, and so on.

- Specifying hexadecimal strings. A hexadecimal string must be in the form 'hhhhhh'x. The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters (0–9, A–F).
- Specifying binary strings. A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of "0"s and "1"s.
- Specifying character strings. For non-numeric types, the value should be enclosed in quotes.
- Specify a variable by specifying &variable_name. A variable is substituted for the value if a matching character, numeric, or tally variable can be located. If a matching variable cannot be found, the string is treated as a literal value. If a numeric comparison is being performed, a character variable is converted to a number - if the conversion fails, the function returns a false result. If a numeric or tally variable is referenced in a character comparison, then the value is the number converted to its display form with leading zeros removed.

duplication

Specify an integer *n* to duplicate the literal value *n* times.

Note: This can only be used for operators that support a single value (for example, Not contains) and where the value is a literal constant and not a substitute variable.

VER Verify the field is composed only of characters specified in the value column.

Example 1

Check the input record and process only those records that contain values of 'Smith' or 'Jones'.

Note: In this case, use operator CU so the contains processing is not case-sensitive.

```
if FLDI(1,,C,'CU','Smith','Jones') then
   return
 else
   return 'DROP'
```

Example 2

Process all records with a salary greater than 75000, where salary is a packed decimal value found at start position 28.

Note: In this case, allow File Manager to calculate the packed decimal field length

```
if FLDI(28,,P,'>',75000) then
   return
 else
   return 'DROP'
```

Example 3

Process all input records with the value 'ABCABCABCABCABC' at start position 10.

Note: The length defaults to 15, the literal value length.

```
If FLDI(10,'=','ABC',5) then
     return
  else
     return 'DROP'
```

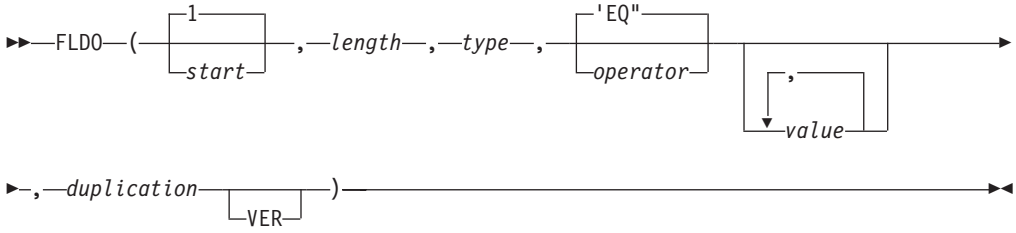
Example 4

The same as Example 2, but using a numeric variable.

```
SETN(salary_high,75000)
if FLDI(28,,P,'>','&salary_high') then
     return
   else
```

FLDO

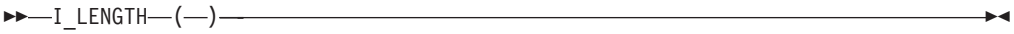
Syntax



FLDO is the same as FLDI except it tests the current output record.

I_LENGTH

Syntax



Can be used in FASTREXX condition expressions.

Returns the length, in bytes, of the input record.

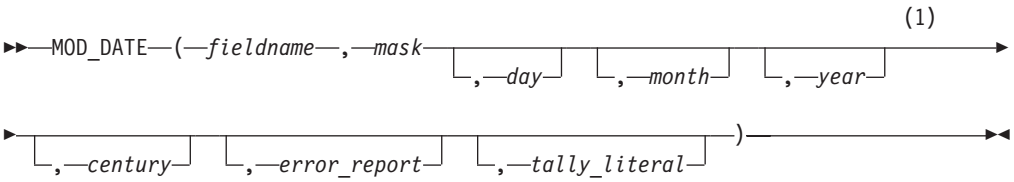
Example 1

Write only records with a length of 100 to the output file DD100.

```
If i_length() = 100 Then
  WRITE('DD100')
```

MOD_DATE

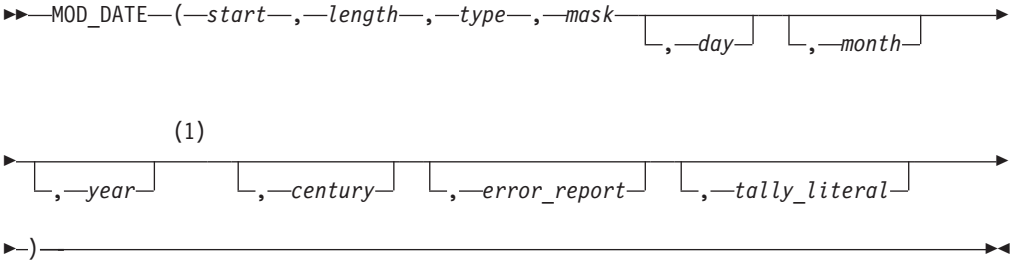
MOD_DATE syntax - with template



Notes:

1 At least one of these parameters (*day*, *month*, *year*) must be specified.

MOD_DATE syntax - without template



Notes:

1 At least one of these parameters (*day*, *month*, *year*) must be specified.

(Can be used in FASTREXX procedures.)

Note:

1. Only modifies valid dates. Input and modified dates must range between and include 15 October 1582 to 31 December 9999.
2. Commas following the last specified argument can be omitted.

MOD_DATE can be run with or without a template and enables you to set, increment, or decrement a date field using year.month and day values. The *fieldname* or location parameters (*start*, *length*, *type*), the *mask* parameter and at least one of the *year* or *month* or *day* values must be specified for the function call to be valid. The input value is taken from the current output buffer and the modified value is stored in the output buffer.

fieldname
A template is required with this form and the name you specify must match a field name in the template. For non-unique names, you can specify a name in the form *groupname.dataname*. Name matching is not case-sensitive. If the name is unqualified, then the first occurrence of the name is used. For dimensioned fields, you can refer to an individual array element by providing a suffix subscript in the form (*nn*), where *nn* is a valid subscript for the dimensioned field. If you do not provide a subscript, the function applies to all elements of the array.

Note:

1. If you are running a copy process and you have specified an input and output template, then the field name must be defined to both input and output templates.
2. If you are copying multiple record layouts, the MOD_DATE function applies to records that have been identified as the record layout which contains this field.

start Position, in bytes, in the input record at which to start reading the field value. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the input record, the function has no effect.

Relative to current INPOS
Can be specified as IPx or INx, or as Px or Nx. If this resolves to a value of less than or equal to zero or greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS
Must be specified as OPx or ONx. If this resolves to a value of less than or equal to zero or greater than the current length of the input record, the function has no effect.

length The length of the field in bytes.

For binary fields, either omit the length or specify 4.

For packed decimal fields, if you specify the length, it must be in the range 1–16. If you omit the length, MOD_DATE attempts to determine the packed field length from the record data and returns only that field.

type The data type of the field. Valid values are:
B Binary.
C Character. This is the default.
P Packed decimal.

mask Picture string describing the date. File Manager supports all pictures described in the *z/OS Language Environment Programming Services*, Appendix B, "Date and time services tables". In addition to those masks, File Manager supports CYYDDD to handle old century value Julian dates. A "C" value of "0" is interpreted as "19", and a "C" value of "1" is

interpreted as "20". By default, 2-digit years lie within the 100-year range starting 80 years prior to the system date covered by the LE masks. The default range is changed by specifying the *century* parameter.

day Day adjustment.

month Month adjustment.

year Year adjustment.

The *day*, *month*, and *year* adjustment values can set, increment, and or decrement the respective value in the date field. The presence of a plus or minus symbol indicates the number provided is to *increment* or *decrement* the current value. The absence of plus or minus symbols indicates the number is to *replace* the current value. If the first non-blank character is an asterisk, then the current day, month, or year value is used to set the current value, otherwise it is derived from the record. For the *month* value or *year* value, you can suffix the value with an "E" to indicate end of month adjustment. If the input date is the end of month, then the resultant date is the end of month, unless day arithmetic is also performed.

Here are examples showing how to code them:
'*+10' Sets the current day, month, or year and adds 10 to it.
'*-10' Sets the current day, month, or year and subtracts 10 from it.
'+10' Adds 10 to the current value found on the record.
'-10' Subtracts 10 from the current value found on the record.
'+3E' Adds 3 months and make adjustments for end of month if required.
'22' Sets the respective day, month, or year value to 22.

Note: If you specify the symbols "*", "+", or "-", you must code the value in quotes.

Month and Year Arithmetic:

If input date is the last day of the month and you specify "E" as the suffix on the month or year value, or if the resulting month has fewer days than the day component of the input date, the result is the last day of the resulting month. Otherwise, the result has the same day component as input date.

Here are some examples:

- Assume today is January 31, 2007. Adding one month results in the end of February, 2007-02-28.
- "+3E" is specified for *month*, and the input date is February 28th 2007. The result is May 31st 2007.
- "+3" is specified for *month*, and the input date is February 28th 2007. The result is May 28th 2007.

century
Specify a value between 0 and 100 to be used to define the 100-year range to interpret a 2-digit year. The default value is 80 years prior to the system date. Use this parameter to adjust that value.

error_report
Specify "Y" if you want to produce error messages when the function cannot modify a date because either the input or resultant date is invalid. The default is to ignore such dates.

tally_literal
Specify a literal to appear on a tally report that counts every successful operation of the function.

Examples

This COBOL copybook describes an input record with different date formats and is used here to illustrate MOD_DATE usage with a template or copybook.

```
01 DATE-REC.
   03 DATE-YYYYMMDD              pic x(8).
   03 DATE-MM-DD-YYYY            pic x(10).
   03 DATE-DD-MM-YYYY            pic x(10).
   03 DATE-YYYYDDD               pic x(7).
   03 DATE-YYYYDDDp              pic 9(7) packed-decimal.
   03 DATE-YYYYMMDDb             pic 9(8) binary.
   03 DATE-CYYDDD                pic 9(6) packed-decimal.
   03 DATE-DDMMYYYY              pic 9(8).
   03 DATE-YYMMDD OCCURS 6 times pic 9(6) packed-decimal.
   03 filler                     pic x.
```

Example 1

Add 60 days to all date fields and report any errors. Note all array fields are modified.

```
$$FILEM DSC INPUT=DDIN,
$$FILEM IGNLEN=YES,
$$FILEM TCIN=hlq.COBOL(SAMPLE),
$$FILEM OUTPUT=DDOUT,PROC=*
MOD_DATE('DATE-YYYYMMDD','YYYYMMDD','+60',,,,Y)
MOD_DATE('DATE-YYYYMMDDB','YYYYMMDD','+60',,,,Y)
MOD_DATE('DATE-MM-DD-YYYY','MM/DD/YYYY','+60',,,,Y)
MOD_DATE('DATE-DD-MM-YYYY','DD/MM/YYYY','+60',,,,Y)
MOD_DATE('DATE-YYYYDDD','YYYYDDD','+60',,,,Y)
MOD_DATE('DATE-YYYYDDDP','YYYYDDD','+60',,,,Y)
MOD_DATE('DATE-CYYDDD','CYYDDD','+60',,,,Y)
MOD_DATE('DATE-DDMMYYYY','DDMMYYYY','+60',,,,Y)
MOD_DATE('DATE-YYMMDD','YYMMDD','+60',,,,Y)
/*
```

Example 2

Same as example 1 without using a copybook.

```
$$FILEM DSC INPUT=DDIN,
$$FILEM OUTPUT=DDOUT,PROC=*
MOD_DATE(1,8,C,'YYYYMMDD','+60',,,,Y)
MOD_DATE(40,4,B,'YYYYMMDD','+60',,,,Y)
MOD_DATE(9,10,C,'MM/DD/YYYY','+60',,,,Y)
MOD_DATE(19,10,C,'DD/MM/YYYY','+60',,,,Y)
MOD_DATE(29,7,C,'YYYYDDD','+60',,,,Y)
MOD_DATE(36,,P,'YYYYDDD','+60',,,,Y)
MOD_DATE(44,,P,'CYYDDD','+60',,,,Y)
MOD_DATE(48,8,C,'DDMMYYYY','+60',,,,Y)
MOD_DATE(56,,P,'YYMMDD','+60',,,,Y)
MOD_DATE(60,,P,'YYMMDD','+60',,,,Y)
MOD_DATE(64,,P,'YYMMDD','+60',,,,Y)
MOD_DATE(68,,P,'YYMMDD','+60',,,,Y)
MOD_DATE(72,,P,'YYMMDD','+60',,,,Y)
MOD_DATE(76,,P,'YYMMDD','+60',,,,Y)
/*
```

Example 3

Add 2 months and 20 days to all date fields and ignore errors.

```
$$FILEM DSC INPUT=DDIN,
$$FILEM IGNLEN=YES,
$$FILEM TCIN=hlq.COBOL(SAMPLE),
```

```
$$FILEM OUTPUT=DDOUT,PROC=*
MOD_DATE('DATE-YYYYMMDD','YYYYMMDD','+20','+2')
MOD_DATE('DATE-YYYYMMDDB','YYYYMMDD','+20','+2')
MOD_DATE('DATE-MM-DD-YYYY','MM/DD/YYYY','+20','+2')
MOD_DATE('DATE-DD-MM-YYYY','DD/MM/YYYY','+20','+2')
MOD_DATE('DATE-YYYYDDD','YYYYDDD','+20','+2')
MOD_DATE('DATE-YYYYDDDP','YYYYDDD','+20','+2')
MOD_DATE('DATE-CYYDDD','CYYDDD','+20','+2')
MOD_DATE('DATE-DDMMYYYY','DDMMYYYY','+20','+2')
MOD_DATE('DATE-YYMMDD','YYMMDD','+20','+2')
/*
```

Example 4

Set all date field values to the current date plus 1 for year, month, and date. Change the century window used for DATE-YYMMDD field to 60.

```
$$FILEM DSC INPUT=DDIN,
$$FILEM IGNLEN=YES,
$$FILEM TCIN=hlq.COBOL(SAMPLE),
$$FILEM OUTPUT=DDOUT,PROC=*
MOD_DATE('DATE-YYYYMMDD','YYYYMMDD','*+1','*+1','*+1')
MOD_DATE('DATE-YYYYMMDDB','YYYYMMDD','*+1','*+1','*+1')
MOD_DATE('DATE-MM-DD-YYYY','MM/DD/YYYY','*+1','*+1','*+1')
MOD_DATE('DATE-DD-MM-YYYY','DD/MM/YYYY','*+1','*+1','*+1')
MOD_DATE('DATE-YYYYDDD','YYYYDDD','*+1','*+1','*+1')
MOD_DATE('DATE-YYYYDDDP','YYYYDDD','*+1','*+1','*+1')
MOD_DATE('DATE-CYYDDD','CYYDDD','*+1','*+1','*+1')
MOD_DATE('DATE-DDMMYYYY','DDMMYYYY','*+1','*+1','*+1')
MOD_DATE('DATE-YYMMDD','YYMMDD','*+1','*+1','*+1','60')
/*
```

NCONTAIN

Syntax

Compares the value represented by *number* against the value or values defined by *match*.

Returns

If the numeric value of any of the *match* arguments is equal to the numeric value of *number*, then NCONTAIN returns 1. Otherwise, NCONTAIN returns 0.

number

The value, represented by a number, a function returning a value or a variable to which a value has been assigned, that you are comparing with *match*.

match The value or list of values that you are comparing with *number*. You can search for up to 20 values at a time.

For a similar function for matching string values, see “CONTAINS” on page 1111. For a FASTREXX-eligible equivalent, see “FLD_CO” on page 1114.

Example 1

If the current record contains a packed decimal value of 10, 20, or 30 starting at column 8, then print the record.

```
If NCO(FLD(8,P),10,20,30) Then PRINT(inrec,'CHAR')
```

Example 2

If the current record contains a zoned decimal value of 11, 12, or 13 starting at column 10, then drop the record from processing.

```
If NCO(FLD(10,5,Z),11,12,13) Then Return 'DROP'
```

O_LENGTH

Syntax

Can be used in FASTREXX condition expressions.

Returns the length, in bytes, of the output record.

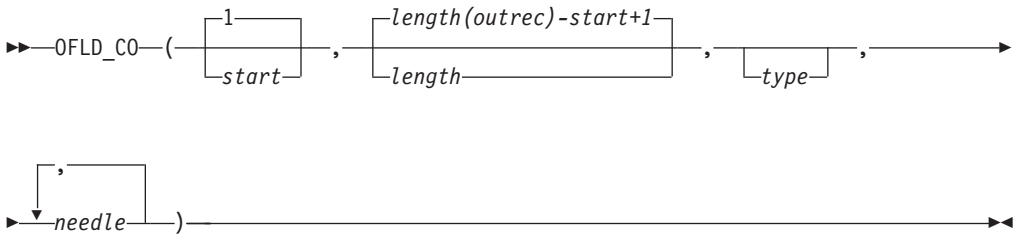
Example 1

If the current length of the output record is 100, overlay the last 20 columns with asterisks.

```
If o_length() = 100 Then
   OVLY_OUT('*',81,20,'*')
```

OFLD_CO

Syntax



(Can be used in FASTREXX condition expressions.)

Searches the field within the output record specified by *start* and *length*, for one or more occurrences of *needle*. On successful execution when searching for a string, also updates the value of OUTPOS to the first byte of the located field in the output record.

Returns

If at least one occurrence of *needle* is found, returns 1. If no occurrences are found, returns 0.

start Position, in bytes, in the output record at which to start searching for occurrences of *needle*. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the output record, the function has no effect.

Relative to current INPOS
Must be specified as IP*x* or IN*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the input record, the function has no effect.

Relative to current OUTPOS
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the output record, the function has no effect.

length Length, in bytes, of the field to be searched.

- For character fields, the length defaults to the remaining record length from the start position to the end of the record (inclusive). A value of 0 also indicates that the field extends to the end of the record.
- For binary fields, you must specify the length. It can be 2, 4, or 8.
- For packed decimal fields, if you specify the length, it must be in the range 1-16. If you omit the length, FLD_CO attempts to determine the packed field length from the record data.
- For zoned decimal fields, if you specify the length, it must be in the range 1-31 or, if the field contains a separate sign character, in the range 1-32. If you omit the length, it defaults to the remainder of the record.

type The data type of the field. Valid values are:

B Binary. FLD_CO interprets binary fields as being signed.

C Character. This is the default. The comparison is case sensitive.

P Packed decimal.

U Interprets the field as Character, but converts it to uppercase before comparing it with *needle*.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

needle String(s) or numeric(s) to search for. For the character types, FLD_CO searches the *haystack* for each of the *needles*. In this context, it behaves like a combination of the FLD and CONTAINS functions. For the numeric data types, the *haystack* is treated as a single numeric field, and an appropriate numeric comparison is performed against each of the *needles*. In this context it behaves like a combination of the FLD and NCONTAIN functions. You can search for up to 20 strings or numerics at a time.

To perform case-insensitive searches, specify *type* as 'U' and *needle* in uppercase.

Example 1

If the current output record contains “MIKE”, “Mike” or “mike” in the first ten columns, then write the record.

```
If FLD_CO(1,10,'U','MIKE') Then WRITE('MDD')
```

Example 2

If the current output record contains “USA”, “Australia” or “England”, then drop the record from processing.

```
If FLD_CO(,,,'USA','Australia','England') Then Return 'DROP'
```

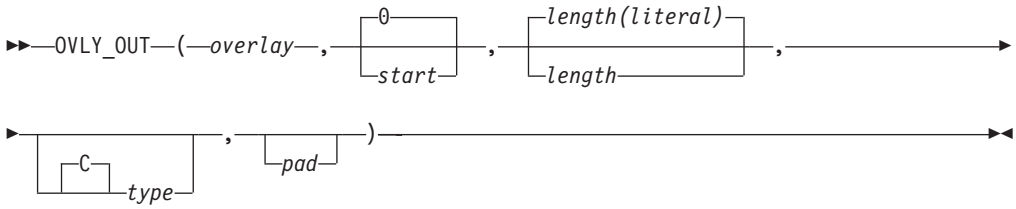
Example 3

If the current output record contains “MIKE”, “Mike” or “mike” in the ten columns immediately after the current INPOS, then write the record.

```
If FLD_CO(P0,10,'U','MIKE') Then WRITE('MDD')
```

OVLY_OUT

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX procedures.

Overlays the output record with a string. If the length of the target field exceeds the length of the literal, the target field is padded to the specified length using the pad character. If the length of the target field is less than the length of the literal, the following occurs:

- Character overlays are truncated on the right without error. For example, OVLY_OUT('ABCD',1,2) overlays 'AB'.
- Numeric overlay truncations are considered to be an error. For example, OVLY_OUT(500000,1,2,'B') fails because you can't fit the specified value into a 2-byte binary field.

On successful execution, also updates the value of OUTPOS to one byte past the end of the field overlaid in the output record.

Returns

A single blank.

overlay An expression that resolves to a string, which is overlaid on that part of the output record specified by *start* and *length*. To be eligible for FASTREXX processing, this must be a literal string, a symbol or a blank-delimited sequence of symbols and/or literal strings.

If the first character of the literal is an ampersand and the literal that follows matches an existing character or numeric variable or tally literal (matching not case-sensitive), then the variable value is substituted according to the type. For example, if the type is character and a numeric or tally value are referenced, then the literal is the numeric value in display format with no leading zeros. If the type is binary, packed or zoned, then the variable value is converted to binary, packed or zoned number.

Note:

1. Conversion errors may occur when converting a character variable to a numeric.
2. If a variable name is not found, then the string is interpreted as a literal.

start Position, in bytes, in the output record at which to start overlaying the string. If you omit *start*, specify zero, or specify a value one greater than the length of the current output record, the field is appended to the end of the output record. If *start* is greater than the current length of the output record, the record is padded with the specified or defaulted pad character from the current record length to the specified start position. Can be specified as:

Absolute position
Must be a non-negative integer that is less than or equal to the maximum length of the output data set. Default value is 0.

Relative to current INPOS
Must be specified as IP*x* or IN*x*. Must resolve to a non-negative integer.

Relative to current OUTPOS
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*. Must resolve to a non-negative integer.

length Length, in bytes, of target field in the output record. Defaults as shown here:

Character fields
Defaults to the length of the literal. A value of 0 indicates that the target field length is the greater of the source (literal) length and the remaining record length. In particular, if 0 is specified for both start and length, then the length of the literal is used as the target length.

Packed decimal
Defaults to the last packed length value determined from the input record by a preceding function. For example,

```
if FLD(1,P) = 2 then
OVLY_OUT('5',1,,P)
```

Uses the length determined by the FLD function to default the packed decimal length. If no previous packed decimal length was calculated, a length error occurs and the procedure is terminated.

type The data type of the literal to be written to the output record.

B Binary. The literal string must represent a positive or negative integer, and is stored in the output field as a signed two's-complement format binary number, right-justified in the target field. The length must be 2, 4, or 8, and cannot be omitted.

C Character. This is the default.

P Packed decimal. The literal string must represent a positive or negative integer, and is stored right-justified in the target field as a signed packed decimal number using the preferred positive ('c'x) and negative ('d'x) sign indicators. The length must be between 1 and 16.

Z Zoned decimal (COBOL external decimal with non-character trailing sign). The literal string must represent a positive or negative integer, and is stored in the output field as a signed zoned decimal number. The length must be between 1 and 31.

pad Pad character. Defaults to the pad character set on the File Manager System Processing Options panel. If the current pad setting is OFF, the default pad character is a blank. For numeric types such as B, P or Z, the pad character is not used when pre-fill characters are required to right-justify a numeric field. The pre-fill characters are always leading zeros, as required by the field type."

See FLD_OUT for a function to overlay the output record with a field from the input record.

Example 1

Set columns one and two of the output record to asterisks.

```
OVLY_OUT('**',1,2)
```

Example 2

Append the two-byte packed decimal value 2 to the end of the output record.

```
OVLY_OUT(2,0,2,P)
```

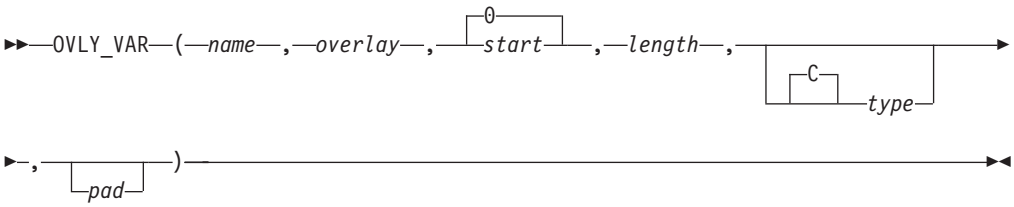
Example 3

Search the input record for a literal and then overlay the last two bytes of that literal with a new literal in the output record.

```
If FLD_CO(1,,C,'AABB') Then
 OVLY_OUT('CC',IP2,2)
```

OVLY_VAR

Syntax



(Can be used in FASTREXX procedures.)

Note: Commas following the last specified argument can be omitted.

Overlays the named character variable with a string. If the length of the target field exceeds the length of the literal, the target field is padded to the specified length using the pad character. If the length of the target field is less than the length of the literal:

- Character overlays are truncated on the right without error. For example, `OVLY_VAR(MYVAR,'ABCD',1,2)` overlays 'AB'.

- Numeric overlay truncations are considered to be an error. For example, `OVLY_VAR(MYVAR,500000,1,2,'B')` fails because you cannot fit the specified value into a 2-byte binary field.

On successful execution, also updates the value of current variable position to one byte past the end of the field overlaid in the variable.

Returns
A single blank.

name 1–256 character variable identifier. Variable name matching is not case sensitive. If the name is not found, a severe error occurs and the procedure is terminated. Cannot be a system character variable or a system numeric variable. See "Using FASTREXX variables" on page 1100.

overlay An expression that resolves to a string, which is overlaid on that part of the variable specified by *start* and *length*. To be eligible for FASTREXX processing, this must be a literal string, a symbol, or a blank-delimited sequence of symbols or literal strings. If the first character of the literal is an ampersand and the literal that follows matches an existing character, numeric variable, or tally literal (matching not case-sensitive), then the variable value is substituted according to the type. For example, if the type is character and a numeric or tally value are referenced, then the literal is the numeric value in display format with no leading zeros. If the type is binary then the variable value is converted to a binary number.

Note: Conversion errors may occur.

start Position, in bytes, in the variable at which to start overlaying the string. If you omit start, specify zero, or specify a value one greater than the length of the current variable, the field is appended to the end of the variable. If start is greater than the current length of the variable, the record is padded with the specified or defaulted pad character from the current variable length to the specified start position. Can be specified as:

Absolute position
Must be a non-negative integer that is less than or equal to the maximum length of the output data set. Default value is 0.

Relative to current variable position
Must be specified as IPx or Inx, OPx or ONx, or as Px or Nx. Must resolve to a non-negative integer.

length Length, in bytes, of target field in the variable. Defaults:

Character fields
To the length of the literal. A value of 0 indicates that the target field length is the greater of the source (literal) length and the remaining variable length. In particular, if 0 is specified for both start and length, then the length of the literal is used as the target length.

Packed decimal
To the last packed length value determined from the input record by a preceding function. For example:

```
if FLD(1,P) = 2 then
OVLY_VAR(MYVAR,'5',1,,P)
```

Uses the length determined by the FLD function to default the packed decimal length. If no previous packed decimal length was calculated a length error occurs and the procedure is terminated.

type The data type of the field. Valid values are:

B Binary. FLD_CO interprets binary fields as being signed.

C Character. This is the default. The comparison is case sensitive.

P Packed decimal.

U Interprets the field as Character, but converts it to uppercase before comparing it with *needle*.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

pad Pad character. Defaults to the pad character set on the File Manager System Processing Options panel. If the current pad setting is OFF, the default pad character is a blank. For numeric types such as B, P, or Z, the pad character is not used when pre-fill characters are required to right-justify a numeric field. The pre-fill characters are always leading zeros, as required by the field type.

Example 1

Set columns one and two of the variable to asterisks.

```
OVLY_VAR(MYVAR,'**',1,2)
```

Example 2

Append the two-byte packed decimal value 2 to the end of the variable.

```
OVLY_VAR(MYVAR,2,0,2,P)
```

Example 3

Search the variable for a literal and then overlay the last two bytes of that literal with a new literal in the variable.

```
If TESTC(MYVAR,'CU','AABB') Then
 OVLY_VAR(MYVAR,'CC',IP2,2)
```

PRINT

Syntax

Prints the *record* string in the specified *format*. The print output destination is determined as follows:

Procedure invocation	Print output destination
From an online panel	Determined by the value of the **PRINTOUT** field on the Set Print Processing Options panel.
From a batch job	SYSPRINT
From REXX program	Determined by the PRINTOUT parameter of the SET function.

Returns

0 Function was successful.

4 Function was unsuccessful and the record is not printed because either:
- A *format* of SNGL or TABL was specified and File Manager cannot determine the record type.
- A template has been specified and the record did not pass the selection criteria.

12 Function was unsuccessful and the record is not printed because either:
- You did not provide parameters.
- A *format* of SNGL or TABL was specified, but no template or copybook was provided

record Any string or variable representing a string can be used, however, INREC or OUTREC are most commonly used in this function.

format Can be CHAR, HEX, SNGL or TABL. If you specify TABL or SNGL format:
- On the function or panel that you are enhancing, you must specify a copybook or template that describes the record type to be printed.
- File Manager determines the type of the record to be printed by comparing its length with the record types in the template, and also by using any record identification criteria in the template.
- When using PRINT with DSC or the Copy Utility (option 3.3) and you have specified both an input and an output copybook or template, then the copybook or template used to format the printed record is determined as follows: if the value of the record to be printed matches the input record (INREC variable), then the input copybook or template is used; otherwise, the output copybook or template is used.

  Please ensure the record value matches the template that File Manager uses to print the data set.
- Only those fields that have been selected in the template are printed.
- If the record has been reformatted by template processing, the variable INREC contains the input record value and OUTREC contains the reformatted output record.

Note: Avoid using PRINT in a REXX procedure for a:
- DSP function
- FCH function
- Print Utility (option 3.2)

because output from PRINT is interspersed with their normal output, which can be confusing. Using PRINT in a REXX procedure that runs from the Find/Change Utility panel (option 3.6), does not result in this problem, because the report produced by the panel is sent to a data set, separate from print output.

Example 1

Print the first hundred records.

```
If PRTCOUNT() < 100 Then PRINT(inrec,'CHAR')
```

Example 2

Print the current input record in TABL format.

```
rc = PRINT(inrec,'TABL')
```

PRTCOUNT

Syntax

Can be used in FASTREXX condition expressions.

Returns the current count of records printed. The count is incremented for each record printed by the DSP function or Print Utility (option 3.2), and for each invocation of the PRINT function.

Example

Print the first 10 input records.

```
If PRTCOUNT() < 10 Then PRINT(inrec,'CHAR')
```

PRT_IN

Syntax

Can be used in FASTREXX condition expressions.

Prints the input record in the format specified.

format
Format in which the input record is to be printed. Valid values are: CHAR, HEX, SNGL, or TABL.

If you specify TABL or SNGL format:

- On the function or panel that you are enhancing, you must specify a copybook or template that describes the record type to be printed.
- File Manager determines the type of the record to be printed by using the record identification criteria if present, or by comparing its length with the record types in the template.
- When using PRT_IN with DSC or the Copy Utility (option 3.3) and you have specified both an input and an output copybook or template, the copybook or template used to format the printed record is the output template.
- Only those fields that have been selected in the template are printed.

count FASTREXX only. The maximum number of times this function will be performed. The default is no limit.

Note: Avoid using PRT_IN in a REXX procedure for:

- DSP function
- FCH function
- Print Utility (option 3.2)

Output from PRT_IN is interspersed with the normal output (from DSP, FCH, or Print) which can be confusing.

Using PRT_IN in a REXX procedure that runs from the Find/Change Utility panel (option 3.6) does not result in this problem because the report produced by the panel is sent to a data set, separate from print output.

Example

Print the first 5 records in hex format and print the remainder in tabular format.

```
PRT_IN('HEX',5)
If recsin() > 5 then PRT_IN('TABL')
```

PRT_OUT

Syntax

Can be used in FASTREXX condition expressions.

Prints the output record in the format specified.

format
Format in which the output record is to be printed. Valid values are: CHAR, HEX, SNGL, or TABL.

If you specify TABL or SNGL format:

- On the function or panel that you are enhancing, you must specify a copybook or template that describes the record type to be printed.
- File Manager determines the type of the record to be printed by using the record identification criteria if present, or by comparing its length with the record types in the template.
- When using PRT_OUT with DSC or the Copy Utility (option 3.3) and you have specified both an input and an output copybook or template, the copybook or template used to format the printed record is the output template.
- Only those fields that have been selected in the template are printed.

count FASTREXX only. The maximum number of times this function will be performed. The default is no limit.

Note: Avoid using PRT_OUT in a REXX procedure for:
- DSP function
- FCH function
- Print Utility (option 3.2)

Output from PRT_OUT is interspersed with the normal output (from DSP, FCH, or Print) which can be confusing.

Using PRT_OUT in a REXX procedure that runs from the Find/Change Utility panel (option 3.6) does not result in this problem because the report produced by the panel is sent to a data set, separate from print output.

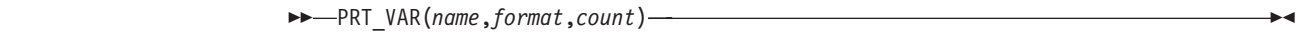
Example

Print the first 5 records in hex format and print the remainder in tabular format.

```
PRT_OUT('HEX',5)
If recsin() > 5 then PRT_OUT('TABL')
```

PRT_VAR

Syntax



Can be used in FASTREXX condition expressions.

Prints the specified variable in the format specified.

name The name of the variable to be printed. This must be the name of an existing character variable.

format

Format in which the output record is to be printed. Valid values are: CHAR, HEX, SNGL, or TABL.

If you specify TABL or SNGL format:

- On the function or panel that you are enhancing, you must specify a copybook or template that describes the record type to be printed.
- File Manager determines the type of the variable (record) to be printed by using the record identification criteria if present, or by comparing its length with the record types in the template.
- When using PRT_VAR with DSC or the Copy Utility (option 3.3) and you have specified both an input and an output copybook or template, then the copybook or template used to format the printed variable is the input template for any variable other than ZOUTREC. ZOUTREC is formatted with the output record.
- Only those fields that have been selected in the template are printed.

count FASTREXX only. The maximum number of times this function will be performed. The default is no limit.

Note: Avoid using PRT_VAR in a REXX procedure for:
- DSP function
- FCH function
- Print Utility (option 3.2)

Output from PRT_VAR is interspersed with the normal output (from DSP, FCH, or Print) which can be confusing.

Using PRT_VAR in a REXX procedure that runs from the Find/Change Utility panel (option 3.6) does not result in this problem because the report produced by the panel is sent to a data set, separate from print output.

Example

Print the first 5 records in hex format and print the remainder in tabular format.

```
PRT_VAR('ZINREC','HEX',5)
If recsin() > 5 then PRT_VAR('ZINREC','TABL')
```

RECCUR (DSEB only)

Syntax

(Can be used in FASTREXX procedures.)

Returns the current record number.

Example

If the current record is the hundredth record in the file, then print it.

```
If RECCUR() = 100 Then PRINT(inrec,'CHAR')
```

RECSIN

Syntax

Can be used in FASTREXX condition expressions.

Returns the count of records read so far from the input data set. When the input data set is a PDS, the RECSIN count restarts for each member being processed.

When used with DSEB, RECSIN returns the record number of the furthest record read so far in the data set. (For example, if you have moved down as far as record number 500 in the data set, then you move up to a previous record, RECSIN still returns 500 after moving up.)

Example

Print every hundredth record.

```
If RECSIN()//100 = 0 Then PRINT(inrec,'CHAR')
```

RECSOUT

Syntax

Notes:

1 When used with a DSC or DSP function, *ddname* is optional.

Can be used in FASTREXX condition expressions.

Returns the count of records so far written to the specified output data set.

The argument you can specify is:

ddname

Specifies that the count of records so far written to the data set identified by the specified *ddname* be returned. If *ddname* is omitted when used with a DSC or DSP function, the default is the ddname of the primary output data set. The primary output data set depends on the File Manager function or panel being used:

Function or panel option	Primary output data set is...
Print Utility (option 3.2)	Determined by the value of the **PRINTOUT** field on the Set Print Processing Options panel. For details, see "Printing from File Manager" on page 299.
DSC function Copy Utility (option 3.3)	The data set that is the target of the copy function.
DSP function	When used in a batch job, the primary output data set is SYSPRINT. When used in a REXX procedure, the primary output data set is determined by the PRINTOUT parameter of the SET function. For details, see "SET (Set Processing Options)" on page 1047.

Specifying the ddname of the primary output data set is the same as omitting the argument.

If you specify a ddname that is not the ddname of the primary output data set and has not previously been specified as the argument to a WRITE function, the value returned is zero.

The count of records written to an output data set is incremented each time a WRITE function is issued against the specified data set. In the case of the primary output data set, the count is also incremented each time a record is written to the data set by the File Manager function. Unless a record is discarded using the RETURN DROP (or STOP IMMEDIATE) instruction, each record selected for processing is written to the primary output data set. For information about how to discard records, see "RETURN return values" on page 1161.

Note: The RECSOUT function treats each member of the primary output data set as separate, that is, the count starts at zero for each output member. However, the count is maintained across members of the input data set so that, if copying from a PDS to a sequential data set, the RECSOUT function reflects the total number of records written, regardless of how many input members are involved.

If you are using the DSC function or Data Copy Utility and you have specified REXX member selection, the RECSOUT function is disabled for the primary output data set. When RECSCOUT targets a *ddname* other than the primary output data set, it still functions as normal. However, you must keep in mind that after a decision has been made to DROP or PROCESS the member, no further records are passed to your REXX procedure, so subsequent records are not counted.

Example 1

If more than one hundred records have been written to the EXT100 file, then terminate File Manager processing.

```
rc = WRITE(EXT100)
If RECSOUT(EXT100) > 100 Then Return 'STOP'
```

RSTR_OUT

Syntax

(Can be used in FASTREXX condition expressions.)

Restores the most recently saved copy of the output buffer.

There are no synchronization restrictions. The SAVE_OUT() invocation that a RSTR_OUT() invocation is "reversing" could have occurred for the current record, or any previously processed record.

There is no stacking of saved output buffers. If RSTR_OUT() is invoked twice in succession, then the second invocation restores the same data as the first.

If RSTR_OUT() is invoked without a prior invocation of SAVE_OUT(), then the output buffer is "cleared". The effect is the same as executing SET_OLEN(0).

SAVE_OUT

Syntax

(Can be used in FASTREXX condition expressions.)

Saves a copy of the current output buffer.

There are no synchronization restrictions. The SAVE_OUT() invocation that a RSTR_OUT() invocation is "reversing" could have occurred for the current record, or any previously processed record.

There is no stacking of saved output buffers. If SAVE_OUT() is invoked twice in succession, then the data saved by the first invocation is lost.

SEGCNT

Syntax

Can only be used in criteria for a segmented template.

Returns the number of previous occurrences of the segment identified by the parameter, *field_name*. The default value of 0 is returned if used with non-segmented processing.

field_name
Must be a valid 01 name in the template being used.

For example, if SEG-A occurs the number of times specified in the header field SEG-NUM-A, the RID criteria shown in Figure 288 can be used to identify SEG-A.

```
⋮
 File Manager    Related Id expression for SEG-A                    Line 1 of 5

 -------- Criteria - Enter 0 to specify related ID expression by field --------
 0 Rid: segcnt('seg-a') < #2                                                  +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
                1  1 SEG-HDR                                    AN      1      4
                2   2 SEG-NUM-A                        9         ZD      1      1
                3   2 SEG-NUM-B                        9         ZD      2      1
                4   2 SEG-NUM-C                        9         ZD      3      1
                5   2 SEG-NUM-D                        9         ZD      4      1
                  ****  End of data  ****
⋮
```

Figure 288. SEGCNT example

SEGLEN

Syntax

```
►►—SEGLEN(field_name)—————————————————————————————————————————————►◄
```

Can only be used in criteria for a segmented template.

Returns the length in the associated field name. This result is used as the segment length for the identified segment if the value is greater than zero.

field_name
: Must be a valid field name for the current layout.

Note:

1. The field name is converted to a full word positive integer.
2. If the field is alphanumeric then it is assumed to contain a binary value. If it is longer than 4 bytes then the rightmost 4 bytes are used to obtain the binary value.
3. The last value returned by the SEGLEN function in successful identification criteria is used as the segment length. This takes precedence over the specification of a segment length field in a template definition.

For example, Consider the excerpt from an XML template with identification criteria for a section of the SMF 30 record SMF30PSS. The length of the section is in the header section SMFRCD30 field SMF30SLN. When the identification criteria for section SMF30PSS is satisfied the segment length is set to the value contained in the header field SMF30SLN.

```
<layout name="SMFRCD30" offset="-4">
  <criteria type="RID" rname="SMF30PSS">
    <exp><![CDATA[
      segcnt('smf30pss') < #SMF30SON &
      segoff(4) >= #SMF30SOF &
```

```
      seglen('SMF30SLN') > 0
    ]]></exp>
  </criteria>
</layout>
```

SEGOFF

```
►►—SEGOFF(adjustment)———————————————————————————————————►◄
```

Can only be used in criteria for a segmented template.

Returns the current offset into the record during segmented identification processing.

adjustment
Default 0. This is a positive or negative number that is added to the offset value returned.

For example, consider the excerpt from an XML template with identification criteria for a section of the SMF 30 record SMF30PSS. The offset to the section is in the header section SMFRCD30 field SMF30SOF. We use related identification criteria with an adjustment of 4 because the header section includes the RDW which File Manager does not include when processing data. The header section is defined with an offset of minus 4 and we use SEGOFF(4) to adjust current offset in File Manager processing by 4 so the returned value can be compared with the offset value contained in the SMF30SOF field.

```
<layout name="SMFRCD30" offset="-4">
  <criteria type="RID" rname="SMF30PSS">
    <exp><![CDATA[
      segcnt('smf30pss') < #SMF30SON &
      segoff(4) >= #SMF30SOF &
      seglen('SMF30SLN') > 0
    ]]></exp>
  </criteria>
</layout>
```

SEGNO

Syntax

```
►►—SEGNO()———————————————————————————————————————————————►◄
```

Can only be used in criteria for a segmented template.

This function has no parameters and returns a numeric value representing the current segment number. It is only relevant when running with a segmented template. The default value of 1 is returned if used with non-segmented processing.

Figure 289 on page 1145 shows header or first segment identification criteria.

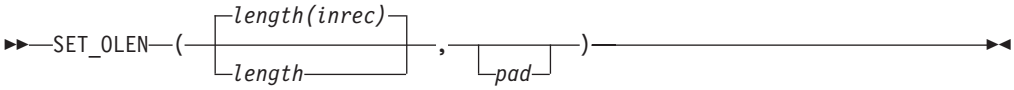
```
⋮
 File Manager                Field Selection/Edit                 Line 1 of 5

 ---Criteria - Enter 0 - Related ID expression, 1 or 2 - expression by field---
 0 Rid: ____________________________________________________________________ +
 1 Id : segno()=1___________________________________________________________ +
 2 Sel: ____________________________________________________________________ +
 Offset ____0          Enter "/" to OR with related ID
 Cmd Seq SHE Ref  Field Name                          Picture  Type Start Length
                  ****  Top of data  ****
                1  1 SEG-HDR                                    AN      1      4
                2   2 SEG-NUM-A                      9          ZD      1      1
                3   2 SEG-NUM-B                      9          ZD      2      1
                4   2 SEG-NUM-C                      9          ZD      3      1
                5   2 SEG-NUM-D                      9          ZD      4      1
                  ****  End of data  ****
⋮
```

Figure 289. SEGNO example

SET_OLEN

Syntax



Note: Commas following the last specified argument can be omitted.

Can be used in FASTREXX procedures.

Sets the length of the output record. If the specified length is greater than the current length of the output record, the pad character is used to fill out the record to the specified length. If the specified length is less than the current OUTPOS, OUTPOS is reset to the new length + 1.

Returns

A single blank.

length Length, in bytes, to set. Must be a non-negative integer. The default value is the length of the input record (which is also the original length of the output record).

pad Pad character. Defaults to the pad character set on the File Manager System Processing Options panel. If the current pad setting is OFF, the default pad character is a blank.

Example

Set the length of the current output record to 80.

```
SET_OLEN(80)
```

SETC

Syntax

Can be used in FASTREXX condition expressions.

Defines or changes a character variable.

name 1–256 character variable identifier. Variable name matching is not case sensitive. Cannot be a system character variable or a system numeric variable. See “Using FASTREXX variables” on page 1100.

value A single value can be entered:

- Specifying hexadecimal strings. A hexadecimal string must be in the form 'hhhhhh'x. The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters (0–9, A–F).
- Specifying binary strings. A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of "0"s and "1"s.
- Specifying character strings. For non-numeric types, the value should be enclosed in quotes.
- Specify a variable by specifying &variable_name. A variable is substituted for the value if a matching character, numeric, or tally variable can be located. If a matching variable cannot be found, the string is treated as a literal value. If a numeric or tally variable is referenced, then the value is the number converted to its display form with leading zeros removed.

  You can refer to a substring of a variable by specifying &*variable_name*(*n*:*length*) which returns a string starting at the *n*th character of the variable for the minimum of the length specified by *length* or the remaining length of variable. If *n* is greater than the length of the variable, the value is treated as null.

duplication

Specify an integer *n* to duplicate the literal value *n* times.

Note: This can only be used where the value is a literal constant and not a substitute variable.

Example 1

Set up a variable called A10 with A value repeated 10 times.

```
If RECSIN() <= 0
SETC ('CHECK_CHAR','A',10)
else
SETC('CHECK_CHAR','B',10)

If FLDI(20,10,,'EQ','&CHECK_CHAR') then
 Return
Else
 Return "DROP"
```

Example 2

Copy the tenth input record into variable REC10.

```
If RECSIN() = 10 then
   SETC('REC10','&ZINREC')
```

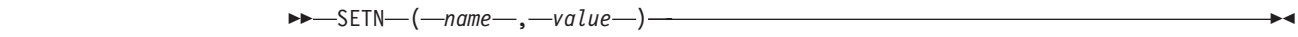
Example 3

Copy the name value from column 3 for 20 on the input record into the variable name.

```
SETC('NAME','&ZINREC(3:20)')
```

SETN

Syntax



Can be used in FASTREXX condition expressions.

Defines or changes a numeric variable.

name 1–256 character variable identifier. Variable name matching is not case sensitive. Cannot be a system character variable or a system numeric variable. See "Using FASTREXX variables" on page 1100.

value An integer or substitute numeric or tally variable preceded with an optional plus or minus sign. The number can set,increment, or decrement the respective value in the variable. The variable is initialized with a zero value if it does not exist. The presence of a plus or minus symbol indicates the number provided is to increment or decrement the current valuerespectively. The absence of plus or minus symbols indicates the number is to replace the current value. Ensure that the value is enclosed in quotes if a plus or minus sign is specified. A substitute variable name should begin with an ampersand (&*variable_name*). This must refer to an existing tally or numeric variable. If the variable does not exist, the procedure fails with a severe error. The maximum number supported is 31 digits plus sign.

Example 1

Count the number of records with surname smith that have an age over 50.

```
If FLDI(1,20,C,'CU','SMITH') and FLD(25,,P) > 50 then
SETN ('Smith_over_50','+1')
```

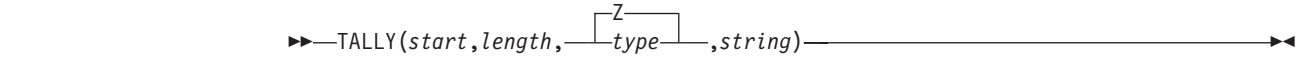
Example 2

Save tally value for first 50 record.

```
TALLY(1,P,'Total Car crashes')
IF RECSIN() = 50 then
   SETN('Total_first_50','&total car crashes')
```

TALLY

Syntax



Can be used in FASTREXX procedures.

Accumulates the value of the specified input record field in a TALLY register and, at the end of the File Manager function, prints on SYSPRINT the TALLY register prefixed by *string*. The TALLY is maintained across members of a PDS.

Note: If REXX member selection has been used with the DSC function or the Data Copy Utility, input records are only passed to the REXX procedure until a decision has been made on whether to DROP or PROCESS the member. TALLY accumulates the value of the specified input record field for all records processed by the REXX procedure. This includes records that were processed for members that were subsequently DROPped, and excludes records that were not passed to the REXX procedure because a decision had already been made to PROCESS or DROP the member.

Returns

A single blank.

start Start, in bytes, position of the field to be tallied in the input record.

length Length, in bytes, of the field to be tallied in the input record.

type The data type of the field to be accumulated. The values that can be specified are:

B Signed binary. If you specify B for *type*, *length* must be 2, 4, or 8.

P Packed decimal. If you specify P for *type*, *length* must be between 1 and 16.

UB Unsigned binary. If you specify UB for *type*, *length* must be 2, 4, or 8.

Z Zoned decimal. This is the default. If you specify Z for *type*, *length* must be between 1 and 32 or, if the field contains a separate sign character or leading blanks, between 1 and 33. TALLY ignores leading blanks for zoned decimal fields, allowing simple character numeric fields to be tallied. Records containing only blanks in the target field are ignored. If a separate sign is present, there must be no blanks between the sign and the zoned data.

string A literal string that is prefixed to the accumulated TALLY total.

The field whose value is to be accumulated starts at position *start* in the input record, and is *length* bytes long. If the sum of *start* and *length* is more than one greater than LENGTH(INREC), the TALLY function returns a blank without changing the TALLY register.

You can code more than one TALLY function in your procedure. File Manager creates a separate TALLY register for each TALLY function with a unique combination of arguments. This means that you can accumulate a given field in more than one TALLY register by specifying a different value for *string* in each TALLY function.

Example

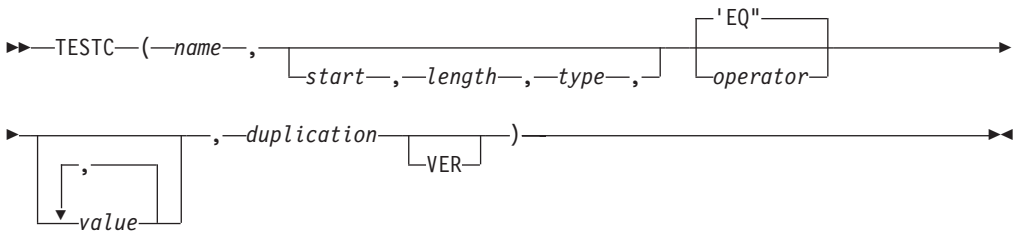
Accumulate hours recorded in personnel records depending on record type.

```
Select
  When (FLD(1,1) == 'E')  Then
     TALLY(15,4,B,'Sum of employee hours')
  When (FLD(1,1) == 'S') Then
```

```
    TALLY(15,4,B,'Sum of supervisor hours')
 Otherwise
    TALLY(28,4,B,'Sum of manager hours')
End
```

TESTC

Syntax



Can be used in FASTREXX condition expressions.

Note:

1. The operator (operator) and non-numeric values should all be enclosed in quotes to avoid syntax errors.
2. The start length and type parameters are optional to provide the capability to examine part of the variable. The second operand can be the operator. See examples provided.
3. If you specify a value for *length* that would cause the record length to be exceeded, a false result is returned.
4. If you specify a numeric type (types B, P, Z), and the specified field contains invalid data for that type, then the function returns a false result. Numeric data is always returned in integer form; that is, the function does not perform scaling of numeric data.

name 1–256 character variable identifier. Variable name matching is not case sensitive. If the name is not found, a variable is created with a length of 1 and value X'00'.

start Position in bytes in the variable at which to start reading the field value. Can be specified as:

Absolute position
Must be a positive integer. If *start* is greater than the current length of the input record, the function has no effect.

Relative start position
Can be specified as IPx or INx, or as Px or Nx, OPx, or ONx. When testing a variable, the current variable position is the basis of calculation. The current variable position is changed by the TESTC function with a contains type operator. If this resolves to a value of less than or equal to zero, the function produces a false result. If this resolves to a value that is greater than the current length of the variable, the function produces a false result.

length The length of the field in bytes.

- For binary fields, you must specify the length. It can be 2, 4, or 8.
- For character fields, if you omit the length, the length is defaulted:

  - For contains type operators, to the rest of the variable.
  - When all values are variable substitutions, to the rest of the variable.
  - Otherwise, the maximum literal value length is be used.
- For packed decimal fields, if you specify the length, it must be in the range 1–16. If you omit the length, the function attempts to determine the packed field length from the variable value and returns only that field.
- For zoned decimal fields, if you specify the length, it must be in the range 1–31 or, if the field contains a separate sign character, in the range 1–32. If you omit the length, the function returns the remainder of the variable. If this exceeds 32, then the function returns a false result.

type The data type of the field. Valid values are:

B Binary. The function interprets binary fields as being signed.

C Character. This is the default.

P Packed decimal.

U Interprets the field as character, but converts it to uppercase before returning the string.

Z Zoned decimal. Interprets all of the COBOL external decimal variants as numeric data.

operator

The default is EQ or =. This function supports all the operators described for dynamic template and criteria edit. For details about the operators supported and their description, see:

- “Dynamic Template panel” on page 529.
- “Record Identification Criteria panel” on page 625.
- “Record Selection Criteria panel” on page 631.

value The value or values entered must be valid in the context of the operator and the field which is being referenced. For example, only certain operators like CO (contains) allow multiple values. Numeric values should be entered when testing numeric fields, and so on.

- Specifying hexadecimal strings. A hexadecimal string must be in the form 'hhhhhh'x. The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters (0–9, A–F).
- Specifying binary strings. A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of "0"s and "1"s.
- Specifying character strings. For non-numeric types, the value should be enclosed in quotes.
- Specify a variable by specifying &variable_name. A variable is substituted for the value if a matching character, numeric, or tally variable can be located. If a matching variable cannot be found, the string is treated as a literal value. If a numeric comparison is being performed, a character variable is converted to a number - if the conversion fails, the function returns a false result. If a numeric or tally variable is referenced in a character comparison, then the value is the number converted to its display form with leading zeros removed.

duplication

Specify an integer *n* to duplicate the literal value *n* times.

Note: This can only be used for operators that support a single value (for example, Not contains) and where the value is a literal constant and not a substitute variable.

VER Verify the field is composed only of characters specified in the value column.

Example 1

SET variable named TESTREC to current input record. Check the variable and process only those records that contain values of 'Smith' or 'Jones'.

Note: In this case, use the operator CU so the contains processing is not case-sensitive. The start, length, and type parameters have been omitted.

```
SETC('TESTREC','&ZINREC')

  if TESTC(TESTREC,'CU','Smith','Jones') then
     return
   else
     return 'DROP'
```

Example 2

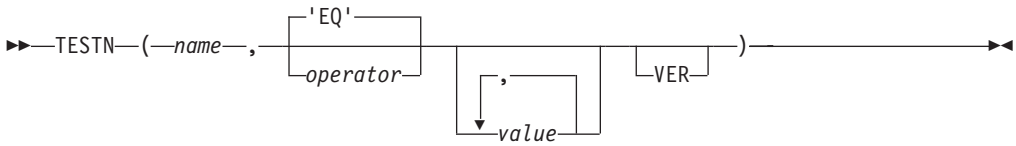
Process all records with a salary greater than 75000, where salary is a packed-decimal value found at start position 28.

Note: In this case, allow File Manager to calculate the packed-decimal field length.

```
SETC('TESTREC','&ZINREC')
  if TESTC(TESREC,28,,P,'>',75000) then
     return
   else
     return 'DROP'
```

TESTN

Syntax



Can be used in FASTREXX condition expressions.

name This is a 1-256 name that matches either a tally literal or a numeric variable. Variable name matching is not case sensitive. If the name is not found, a numeric variable is created with value of 0.

operator

The default is EQ or =. This function supports all the operators described for dynamic template and criteria edit. For details about the operators supported and their description, see:

- "Dynamic Template panel" on page 529.
- "Record Identification Criteria panel" on page 625.
- "Record Selection Criteria panel" on page 631.

value The value or values entered must be valid in the context of the operator

and the field which is being referenced. For example, only certain operators like CO (contains) allow multiple values. Numeric values should be entered when testing numeric fields, and so on.

- Specifying hexadecimal strings. A hexadecimal string must be in the form 'hhhhhh'x. The value enclosed in quotes must be an even number of characters and contain valid hexadecimal characters (0–9, A–F).
- Specifying binary strings. A binary string must be in the form 'nnnnnn'b. The value enclosed in quotes must be a combination of "0"s and "1"s.
- Specifying character strings. For non-numeric types, the value should be enclosed in quotes.
- Specify a variable by specifying &variable_name. A variable is substituted for the value if a matching character, numeric, or tally variable can be located. If a matching variable cannot be found, the string is treated as a literal value. If a numeric comparison is being performed, a character variable is converted to a number - if the conversion fails, the function returns a false result. If a numeric or tally variable is referenced in a character comparison, then the value is the number converted to its display form with leading zeros removed.

VER Verify the field is composed only of characters specified in the value column.

Example 1

Count the number of records with A in the first byte and stop processing after 20.

```
IF FLD(1,1) = 'A' then
  SETN(COUNTA,'+1')

if TESTN(COUNTA,'>',20) then
     return "STOP IMMEDIATE"
   else
     return
```

Example 2

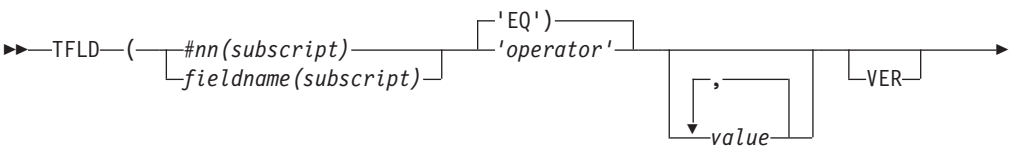
When the total of packed decimal field start at column 28 is greater than 100, stop processing

Note: In this case, allow File Manager to calculate the packed-decimal field length.

```
TALLY(28,P,'Total Sales')
  if TESTN('Total Sales',,'>',100) then
     return "STOP IMMEDIATE"
   else
     return
```

TFLD

Syntax



(Can be used in FASTREXX condition expressions.)

Performs a conditional test against any field defined in a template. For dimensioned fields, you can apply the condition to any or all of the elements of the array. You can only use this function if the associated function is running with a copybook or template.

Note: The Field name (*fieldname*), field reference (#*nn*), operator (*operator*), and non-numeric values should all be enclosed in quotes to avoid syntax errors.

#*nn* **or** *fieldname*

#*nn* Use this form when providing free-format criteria during template edit. *nn* is the field reference number displayed during template edit. It is not valid to use *fieldname* for a field reference when providing criteria during template edit.

fieldname
Use this form of identifying fields when coding user procedures. For non-unique names, you can specify a name in the form *groupname.dataname*. Name matching is not case-sensitive. If the name is unqualified, then the first occurrence of the name is used. Do not code #*nn* values in user procedures, as the displayed field reference values do not identify the correct field when running from a user procedure.

subscript
This applies *only* to dimensioned fields. You can specify one of these forms:

(ANY) This is the default if you do not specify a subscript for a dimensioned field and it indicates that at least one element of the associated array must satisfy the condition for a true result.

(ALL) This indicates that all elements of the associated array must satisfy the condition for a true result.

(*nn***)** This refers to a single array element and you should provide a valid subscript for the dimensioned field.

operator
The default is EQ or =. This function supports all the operators described for dynamic template and criteria edit. For details about the operators supported and their description, see
- "Dynamic Template panel" on page 529
- "Record Identification Criteria panel" on page 625
- "Record Selection Criteria panel" on page 631

value The value or values entered must be valid in the context of the operator and the field which is being referenced. For example, only certain operators like CO (contains) allow multiple values. Numeric values should be entered when testing numeric fields, and so on.

Specifying hexadecimal strings
A hexadecimal string must be in the form `'hhhhhh'x`. The value

enclosed in quotes must be an even number of characters and contain valid hexadecimal characters (0–9, A–F).

Specifying binary strings
A binary string must be in the form '*nnnnnn*'b. The value enclosed in quotes must be a combination of "0"s and "1"s.

Specifying character strings
For non-numeric types, the value should be enclosed in quotes.

VER Verify the field is composed only of characters specified in the value column.

Example 1

Check every element of the dimensioned field CONTACTS and process only those records with contact values of 'Smith' or 'Jones'.

Note: In this case, we use operator CU so the contains processing is not case-sensitive.

```
if TFLD('CONTACTS(ANY)','CU','Smith','Jones') then
   return
 else
   return 'DROP'
```

Example 2

Check the monthly pay for a contract record types and process those with every month occurrance higher than 8000.

Note: MPAY is not unique, so we qualify which MPAY we want to check.

```
Copybook
 01 REC-CONTRACT.
    05 MPAY      PIC S9(8) Binary OCCURS 12 Times.
 01 REC-Employee.
    05 MPAY      PIC S9(8) Binary OCCURS 12 Times.

 if TFLD('REC-CONTRACT.MPAY(ALL)','>',8000) then
   return
 else
   return 'DROP'
```

Performance notes

TFLD is faster than FLD_CO, but requires a template to reference a field value.

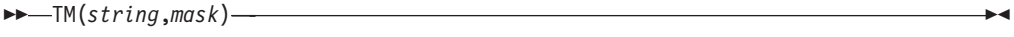
```
TFLD('#3',,'CO','A')  & TFLD('#3',,'CO','B')
```

would be faster if coded as:

```
TFLD('#3',,'ACO','A','B')
```

TM

Syntax



Tests selected bits of a string and sets the condition code accordingly. See “FLD_TM” on page 1118 for a similar function that is FASTREXX eligible.

Returns

If the tested bits are all ones, then TM returns 1. Otherwise, TM returns 0.

string A literal string, or variable representing a string.

mask A bit-string determining which bits to test in *string*.

The length of the test is based on the length of the shorter of the two arguments, *string* and *mask*. A *mask* bit of one indicates that the equivalent bit in *string* is to be tested. When a *mask* bit is zero, the equivalent *string* bit is ignored.

Example 1

Test the third byte of the input record and, if the low order bit is set, overlay a hex FF into the second byte of that record.

```
If TM(FLD(3,1),'01'x) Then Do
  outrec = OVERLAY('FF'x,outrec,2)
  Return
 End
Return 'DROP'
```

Example 2

Test the third byte of the input record and, if the high order bit is set, logically OR a hex 04 over the contents of the second byte of that record.

```
If TM(FLC(3,1),'10000000'b) Then Do
  outrec = OVERLAY(BITOR(fld(2,1),'04'x),outrec,2)
  Return
End
Return 'DROP'
```

TOP (DSEB only)

Syntax

(Can be used in FASTREXX procedures.)

Moves to the first input record.

UP (DSEB only)

Syntax

(Can be used in FASTREXX procedures.)

Moves up *n* number of input records, or to the first input record, if there are less than *n* records above the current input record.

If, after moving, the current input record is the first input record, then the UP function returns the string value “TOF” (top of file).

UPDATE (DSEB only)

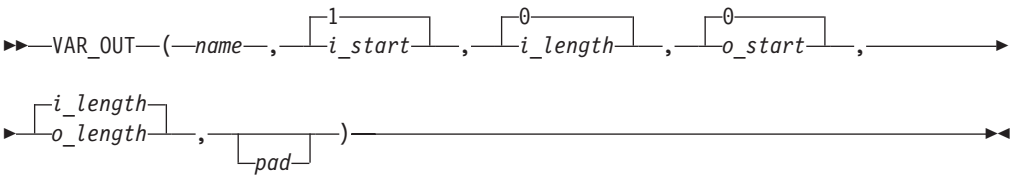
Syntax

(Can be used in FASTREXX procedures.)

Replaces the current input record with the value in OUTREC. If you leave DSEB or move to another record before calling the UPDATE function, then any changes you made to the current OUTREC are lost.

VAR_OUT

Syntax



(Can be used in FASTREXX procedures.)

Note: Commas following the last specified argument can be omitted.

Overlays the output record with a field from the variable. See "OVLY_OUT" on page 1131 for a function to overlay the output record with a literal. If the target field length exceeds the source field length, then the source field is padded to the specified length using the *pad* character. If the target field length is less than the source field, the source field is truncated from the right. On successful execution, also updates the value of OUTPOS to one byte past the end of the field overlaid in the output record.

Returns
A single blank.

name A 1–256 character variable identifier. Variable name matching is not case-sensitive. If the name is not found, a variable is created and populated from the current input record.

i_start Position, in bytes, in the variable at which to start reading the field to be copied. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*, or as OP*x* or ON*x*. Must resolve to a positive integer.

i_length
Length, in bytes, of the source field. Must be a non- negative integer. Defaults to 0. If you omit *i_length* or specify zero, the remainder of the

variable from the *i_start* position is used. This also applies if you specify a value that would cause the source field to be read from beyond the end of the current variable.

o_start Position, in bytes, in the output record at which to start overlaying the copied field. If you omit *o_start* or specify zero, the field is appended to the end of the output record. If *o_start* is greater than the current length of the output record, the record is padded with the specified or defaulted *pad* character from the current record length to the specified *start* position. Can be specified as:

Absolute position
Must be a positive integer. Default value is 1.

Relative to current INPOS
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*, or as OP*x* or ON*x*. Must resolve to a positive integer.

Relative to current OUTPOS
Can be specified as OP*x* or ON*x*, or as P*x* or N*x*. Must resolve to a positive integer.

o_length
Length, in bytes, of the target field. Defaults to the source field length (*i_length*). A value of 0 indicates that the target field length is the greater of *i_length* and the remaining output record length. If 0 is specified for both *o_start* and *o_length*, then *i_length* is used as the target length.

pad Pad character. Defaults to the pad character set on the File Manager System Processing Options panel (when processing online) or the pad character specified in the SET function (when running in batch). If the current pad setting is OFF or unspecified, the default pad character is a blank.

Example 1

Copy the characters in columns 1 and 2 of the variable to columns 3 and 4 of the output record.

```
VAR_OUT(MYVAR,1,2,3,2)
```

Example 2

Append the characters in columns 11 and 12 of the variable to the end of the output record, padded with two blanks.

```
VAR_OUT(MYVAR,11,2,0,4,' ')
```

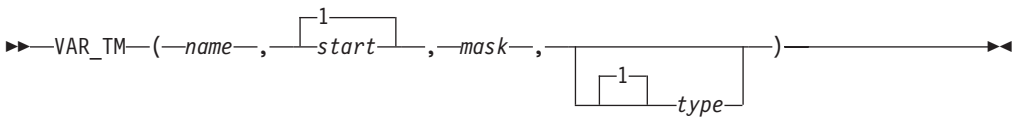
Example 3

Search a field in the variable for the characters 'AA' and, if found, copy to the end of the output record (assumes that OUTPOS is still set to end of output record).

```
IF TESTC(MYVAR,'CU','AA') Then
/* when successful, updates variable position  to 12 */
 VAR_OUT(MYVAR,N2,2,P0,2)
/* copies from variable position - 2, appends to end of output record */
```

VAR_TM

Syntax



(Can be used in FASTREXX procedures.)

Note: Commas following the last specified argument can be omitted.

name 1–256 character variable identifier. Variable name matching is not case sensitive. If the name is not found, the function returns a false result.

start Position, in bytes, in the variable at which to start testing. The length of the field is defined by the mask (*mask*). Can be specified as:

Absolute position
Must be a positive integer. Default value is 1. If *start* is greater than the current length of the variable, the function has no effect.

Relative to variable position
Can be specified as IP*x* or IN*x*, or as P*x* or N*x*, or as OP*x* or ON*x*. If this resolves to a value of less than or equal to zero, the function results in an error. If this resolves to a value that is greater than the current length of the variable, the function has no effect.

mask A bit string determining which bits to test in the field. The length of *mask* defines the length of the input field. This field defines a bit string mapping used to test the specified bits in the variable. You can use the bit string, hex string or character string formats to define this field, Therefore, '0100 0000'b, '40'x, and ' ' are all legitimate and equivalent ways of defining a mask to test the second bit of a one-byte field.

type The type of test:

1 VAR_TM returns True (1) if all the bits that are on in the mask are on in the variable field. This is the default value.

0 VAR_TM returns True (1) if all the bits that are on in the mask are off in the variable field.

M VAR_TM returns True (1) if at least one of the bits that are on in the mask is on in the variable, and at least one is off.

N VAR_TM returns True (1) if at least one of the bits that are on in the mask is off in the variable field.

Example 1

Test the third byte of the variable and, if the low order bit is set, overlay a hex FF into the second byte of the output record.

```
If VAR_TM(3,'01'x) Then Do
  OVLY_OUT('ff'x,2,1)
  Return
End
Return 'DROP'
```

Example 2

Test the third byte of the variable and, if some of the three high order bits are set, and some are not, overlay the contents of the second byte of that record with a hex 04.

```
If VAR_TM(3,'11100000'b,M) Then Do
  OVLY_OUT('04'x,2,1)
```

Example 3

Test the current position of the variable and, if the low order bit is set, overlay a hex FF into the byte prior to this location in the output record.

```
If VAR_TM(P0,'01'x) Then Do
  OVLY_OUT('ff'x,IN1,1)
  Return
End
Return 'DROP'
```

WRITE

Syntax

Notes:

1 When used with a DSC or DSP function, *ddname* is optional.

Can be used in FASTREXX procedures.

Writes a record to the specified data set or sets.

Returns
A single blank.

ddname
Specifies a record is to be written to the data set identified by the specified *ddname*. If *ddname* is omitted when used with a DSC or DSP function, the default is the ddname of the primary output data set. The primary output data set depends on the File Manager function or panel being used:

Function or panel option	Primary output data set is...
Print Utility (option 3.2)	Determined by the value of the **PRINTOUT** field on the Set Print Processing Options panel. For details, see "Set Print Processing Options panel (option 0.1)" on page 658.
Copy Utility (option 3.3) or DSC function	The data set that is the target of the copy function. For the DSC function, if a ddname is associated with the primary output data set, then specifying that ddname is the same as omitting the argument. Do not target the primary output data set with another ddname.

Function or panel option	Primary output data set is...
DSP function	When used in a batch job, the primary output data set is SYSPRINT. When used in a REXX procedure, the primary output data set is determined by the PRINTOUT parameter of the SET function. For details, see "SET (Set Processing Options)" on page 1047. If a ddname is associated with the primary output data set, then specifying that ddname is the same as omitting the argument. Do not target the primary output data set with another ddname.

Except for the primary output data set, the data set attributes of the output data set are derived from:

- The pre-allocated data set DCB attributes. Existing record formats, lengths and block sizes are preserved.
- If it has been allocated without DCB attributes then these attributes are inherited from the input data set.

Sequential data sets specified on a WRITE statement to which no records have been written during the execution of a program are still opened and closed by File Manager. This means that:

- Newly allocated data sets with DISP=(NEW,...) are initialized to an "empty" state and only contain an EOF record.
- Existing data sets with data are handled as follows:
  - DISP=(OLD,...) data sets are reset to an "empty" state and contain only an EOF record; all previously existing data is lost.
  - DISP=(MOD,...) data sets have their data preserved unaffected.

The record is written from the contents of one of the following REXX variables:

- If it has been assigned a value by the procedure, the variable OUTREC.*ddname*, where *ddname* is the ddname specified in the WRITE function.
- If the variable OUTREC.*ddname* has not been assigned a value, or has been unassigned using the DROP instruction, the File Manager-defined variable, OUTREC.

The length of the record written depends upon the data set attributes of the output data set. If the output data set contains variable-length records, the length of the record is determined from the length of the data in the REXX variable. If the length of the data is greater than the maximum record length specified in the data set attributes, the record is truncated. If the output data set contains fixed-length records, the length of the record written is the length specified in the data set attributes, truncated or padded as necessary.

If the record format of the output data set specifies that the records contain a carriage control character, depending on the output device, the first character of the record data in the REXX variable is interpreted as a carriage control character. For more information about records containing carriage control characters, see the *z/OS DFSORT Installation and Customization*.

If you are using the DSC function or Data Copy Utility and the input data set is a PDS(E) and the *ddname* of the target refers to a PDS(E), then members are created in the target data set with names matching the name of the input data set member as the result of the WRITE execution.

Example 1

If the current record is type 01, then write it to the DD01 file.

```
If FLD(1,2) = 01 Then WRITE('DD01')
```

Example 2

If the current record is type 02, then write it to the DD02 and DD02COPY files.

```
If FLD(1,2) = 02 Then WRITE('DD02','DD02COPY')
```

RETURN return values

In REXX, you can use the RETURN instruction to leave a procedure. You can optionally specify a character string as a parameter on the RETURN instruction. This character string is returned to the caller of the procedure. File Manager recognizes and processes the following character strings if specified on the RETURN instruction. The character strings can be specified in uppercase, lower case, or a mixture of both.

DROP

The character string DROP tells File Manager to not write the current record to the primary output data set. The effect of specifying DROP on the RETURN instruction depends upon the File Manager function or panel being used:

- **For DSC or Copy Utility (option 3.3)**, the current record is not copied to the target data set.
- **For DSP or Print Utility (option 3.2)**, the current record is not printed.
- **For DSU**, the current record is not updated, regardless of any changes that have been made to the data in the variable OUTREC.
- **For DSEB**, the current record is not updated, unless the UPDATE function has already been called for the current record.

If this string is returned by a REXX procedure run in conjunction with the DSC function or Data Copy Utility and REXX member selection has been specified, it is treated as if it was a RETURN with no argument strings and a warning message is issued. Subsequent records continue to be passed to the REXX procedure until a decision has been made on whether to DROP or PROCESS the member.

Here are some examples:

```
RETURN 'DROP'
Return 'Drop'
Return drop  /* assumes no variable named 'drop' has been assigned */
```

DROP MEMBER

Indicates that the member is to be excluded from the copy. No more records in this member are passed to the REXX procedure for testing. Processing continues with the next member.

This string is only valid when used in conjunction with the DSC function or Data Copy Utility, and REXX member selection has been specified. If REXX member

selection has NOT been specified, it is treated as a RETURN with no argument strings, and a warning message is issued. Subsequent records continue to pass to the REXX procedure.

PROCESS MEMBER

Indicates that the member is to be included in the copy. No more records in this member are passed to the REXX procedure for testing. The member is copied intact, subject to any specified template processing, which is performed before the user REXX procedure is invoked. Processing continues with the next member.

This string is only valid when used in conjunction with the DSC function or Data Copy Utility, and REXX member selection has been specified. If REXX member selection has NOT been specified, it is treated as a RETURN with no argument strings, and a warning message is issued. Subsequent records continue to pass to the REXX procedure.

STOP

The character string STOP tells File Manager to terminate the current File Manager function after the current record has been written to the primary output data set. The effect of specifying STOP on the RETURN instruction depends upon the File Manager function or panel being used:

- **For DSC or Copy Utility (option 3.3)**, the current record is copied to the target data set and the function is terminated.
- **For DSP or Print Utility (option 3.2)**, the current record is printed and the function is terminated.
- **For DSU**, the current record is updated if the data in the variable OUTREC has been changed, and the function is terminated.
- **For DSEB**, the current record is not updated, unless the UPDATE function has been called for the current record. In either case, the function is terminated.

If this string is returned by a REXX procedure run in conjunction with the DSC function or Data Copy Utility and where REXX member selection has been specified, it is treated as if it was a RETURN with no argument strings and a warning message is issued. Subsequent records continue to be passed to the REXX procedure until a decision has been made on whether to DROP or PROCESS the member.

Here are some examples:

```
RETURN 'STOP'
Return 'Stop'
Return stop  /* assumes no variable named 'stop' has been assigned */
```

STOP IMMEDIATE

The character string STOP IMMEDIATE tells File Manager to terminate the current File Manager function without writing the current record to the primary output data set. The effect of specifying STOP IMMEDIATE on the RETURN instruction depends upon the File Manager function being used:

- For DSC (Data Set Copy), the current record is not copied to the target data set and the function is terminated.
- For DSP (Data Set Print), the current record is not printed and the function is terminated.

- For DSU (Data Set Update), the current record is not updated, regardless of any changes that have been made to the data in the variable OUTREC, and the function is terminated.
- For DSEB (Data Set Edit Batch), the current record is not updated, unless the UPDATE function has been called for the current record. In either case, the function is terminated.

If this string is returned by a REXX procedure run in conjunction with the DSC function or Data Copy Utility and where REXX member selection has been specified, it is treated as if it was a RETURN with no argument strings and a warning message is issued. Subsequent records continue to be passed to the REXX procedure until a decision has been made on whether to DROP or PROCESS the member.

Here are some examples:

```
RETURN 'STOP IMMEDIATE'
Return 'Stop Immediate'
Return stop immediate  /* assumes variables 'stop' and 'immediate' not assigned */
```

Part 3. Appendixes

Appendix A. File Manager messages

Messages

Messages issued by "base" File Manager have a unique alphanumeric identifier with the format:

FMNB*annn*

where:

a Is an alphabetic character.
nnn Is a 3-digit number.

When the message is issued by a batch function, the identifier is printed in front of the text of the message.

When the message is issued by an online function, the identifier is not displayed with the message. However, you can obtain the identifier of a message by typing MSGID on the command line and pressing Enter. This command causes the message identifier of the last message issued to be displayed on your screen.

Many of the online messages are comprised of a short message and long message. The short message is displayed in the upper right-hand corner of the panel. The long message is only displayed if you press the F1 key when the short message is displayed.

In this section, selected batch messages are listed in alphanumeric order. For each message, the information provided comprises:

- The message identifier.
- The text of the message.
- An explanation of the message.
- The required user response.

FMNBA009 Insufficient virtual storage available

Explanation: File Manager does not have enough virtual storage to start or to continue the current function.

User response: Run File Manager with a larger region size.

FMNBA011 Function *name* is not supported in batch mode

Explanation: You called a File Manager function that cannot be run in batch mode.

User response: Use a different function, or invoke File Manager in a different mode.

FMNBA012 Function *name* is not included in File Manager for z/OS

Explanation: You called a function that is not part of File Manager for z/OS.

FMNBA013 Function *name* is not supported in full-screen mode

Explanation: You called a File Manager function that cannot be run in full-screen mode.

User response: Use a different function, or invoke File Manager in a different mode.

FMNBA014 Function *name* is not supported in command mode

Explanation: You called a File Manager function that cannot be run in command mode.

User response: Use a different function, or invoke File Manager in a different mode.

FMNBA015 Function *name* is not supported in line mode

Explanation: You called a File Manager function that cannot be run in line mode.

User response: Use a different function, or invoke File Manager in a different mode.

FMNBA016 Function *name* is not supported in z/OS

Explanation: You called a File Manager function that is not available.

User response: For a list of functions, refer to Chapter 12, "Introduction to programming with File Manager functions," on page 397.

FMNBA031 File Manager security setup failed - RACROUTE R15 X'*nn*', RC X'*nn*', RSCD X'*nn*',

Explanation: File Manager is unable to get RACF information via RACROUTE. For more information, refer to the RACROUTE return and reason codes.

User response: Ensure that your RACF environment is set up correctly. Refer to your RACF documentation, the *File Manager Program Directory*, and the *File Manager Customization Guide*.

FMNBA032 File Manager security setup failed - module 'FMNSECUR' not found in LPA

Explanation: The security exit must exist in LPA.

User response: Refer to the *File Manager Program Directory* and the *File Manager Customization Guide*.

FMNBA033 Not authorized for *function*

Explanation: Your user ID is not authorized to use the function that you specified.

User response: If you need to use this function, contact your system support to enable your user ID for this function or function group. Refer to the *File Manager Customization Guide*.

FMNBA036 Not authorized for fullpack access to VOLSER *volser*

Explanation: Your user ID is not authorized to use the fullpack disk functions on the specified volser.

User response: If you need to use fullpack disk functions, contact your system support to enable your user ID for fullpack access as described under "Customizing the Security Environment" in the *File Manager Customization Guide*.

FMNBA037 Function *name* not supported in authorized mode

Explanation: You invoked a function that cannot be run in authorized mode.

User response: Invoke File Manager in unauthorized mode.

FMNBA038 VOLSER *xxxxxx* too large for fullpack access

Explanation: The volume *xxxxxx* is too large for fullpack disk functions. Fullpack disk functions are restricted to volumes with total capacity less than 65536 tracks.

User response: Use a smaller disk

FMNBA040 Panel display error *rc* for panel *name*

Explanation: The panel cannot be displayed.

User response: Save any system error information and contact your system support.

FMNBA055 WRITE(&dd) issued that would cause a re-open of a file with UNIT=AFF specified

Explanation: The DD name referenced has been closed due to processing of another WRITE function that shares the same tape unit. This file cannot be re-opened once it has been closed.

User response: Examine the logic in your procedure and correct the logic so that you are not writing to one file then the next file and then the previous file again.

FMNBA063 VSAM Return code=*rc* Error code=*error code*

Explanation: The requested CICS I/O request failed against the selected CICS resource. VSAM returned the listed return code and error code. Additional error messages may have been written to the CICS or system log.

User response: Look up the VSAM return and error codes in *DFSMS Macro Instructions for Data Sets* for further information.

FMNBA065 File Manager profile save failed

Explanation: File Manager was updating the profile and an error occurred.

User response: Check the FMNPROF allocation or the FMNPROF specification in the installation profile. Refer to the *File Manager Customization Guide*.

FMNBA066 File Manager profile not found or in error, defaults used

Explanation: The File Manager installation or user profile was not found or is in error. The default values supplied by IBM are used.

User response: Refer to the *File Manager Customization Guide*.

FMNBA067 File Manager profile parameter *parameter*=*value* unknown, defaults used

Explanation: An invalid parameter was found in the File Manager installation profile. The default values supplied by IBM are used.

User response: Refer to the *File Manager Customization Guide*.

FMNBA068 Profile error on or near '*parameter*', File Manager defaults used

Explanation: A syntax error was detected in the File Manager installation profile, in or near the indicated parameter. The job continues using the IBM-supplied defaults.

User response: Correct the user-supplied File Manager profile. Refer to the *File Manager Customization Guide*.

FMNBA075 Required HOSTNAME parameter not specified.

Explanation: A client on a system that supports IPv6 attempted to establish a connection with File Manager but did not pass the required HOSTNAME parameter.

User response: Contact your system support.

FMNBA076 IPV is not at the minimum required service level.

Explanation: IPV is not at the minimum required service level to interface with your installed version of File Manager/CICS.

User response: Upgrade IPV to the required service level and then rerun File Manager.

FMNBA083 RECLIMIT *nnnnn* start value exceeds *nnnnn* record length

Explanation: The SET RECLIMIT processing option specifies a start position that is greater than the record length of the record you are trying to print.

User response: Use the SET function and set the RECLIMIT start to a position within the record, then rerun the function.

FMNBA084 REXX not available, printout routed to SYSPRINT

Explanation: You specified SET PRINTOUT=REXX, but File Manager was not called from a REXX procedure. The print output is routed to SYSPRINT instead. (SET PRINTOUT=SYSPRINT).

User response: Use SET PRINTOUT=REXX only when calling File Manager from a REXX procedure.

FMNBA091 Missing or invalid control card

Explanation: The message indicates an error in a control card. A subsequent message contains more information about the error.

User response: Correct the control card and rerun the job.

FMNBA092 Parameter *parameter* is missing

Explanation: You omitted a required parameter.

User response: Provide all required parameters.

FMNBA093 Missing or invalid function code

Explanation: You either omitted a function code or specified an invalid function code.

User response: Provide the correct function code in the control statement.

FMNBA094 Invalid syntax near card column *nn*

Explanation: The syntax of the control statement is invalid. A scale is printed to help you find the error.

User response: Correct the control statement.

FMNBA095 Too many parameters

Explanation: You might have specified excessive or duplicate parameters.

User response: Remove excessive or duplicate parameters.

FMNBA096 Parameters *parameter1* and *parameter2* are mutually exclusive

Explanation: You specified two parameters that cannot be used together.

User response: Provide the correct parameters.

FMNBA097 Parameter *parameter1* or *parameter2* missing

Explanation: You did not supply a required parameter.

User response: Specify one of the indicated parameters.

FMNBA098 Parameter *parameter1* or *parameter2* or *parameter3* missing

Explanation: You did not supply a required parameter.

User response: Specify one of the indicated parameters.

FMNBA099 Invalid continuation, syntax error near card column *nn*

Explanation: An expected continuation of the control statement could not be found or contains invalid syntax.

User response: Either remove the continuation indicator or correct the continuation card.

FMNBA100 Unexpected end of parameter specification

Explanation: The parameters for the File Manager invocation are terminated by a comma.

User response: Correct the parameter specification for File Manager invocation.

FMNBA101 IMS function requested and IMS component is not available

Explanation: The function you have tried to use is a File Manager IMS function, but the load of the File Manager IMS load module, FMNMOD1, failed. This module might have been deleted or protected (for example, if the functions supported by this module are not needed at your installation).

User response: If you need to use this function, contact your system support.

FMNBA102 DB2 function requested and DB2 component is not available

Explanation: The function you have tried to use is a File Manager DB2 function, but the load of the File Manager DB2 load module, FMNMOD2, failed. This module might have been deleted or protected (for example, if the functions supported by this module are not needed at your installation).

User response: If you need to use this function, contact your system support.

FMNBA113 Parameter *parameter* invalid or not applicable for this function

Explanation: You specified an unknown parameter or a parameter that is not applicable for this function.

User response: Correct the File Manager invocation or the control statement.

FMNBA114 Parameter *parameter* is ambiguous

Explanation: File Manager cannot determine an abbreviated parameter that you specified.

User response: Specify the full parameter name.

FMNBA115 Value of parameter *parameter* missing

Explanation: A value for the specified parameter could not be found.

User response: Correct the File Manager invocation or the control statement.

FMNBA116 Extraneous parameter *parameter* ignored

Explanation: You supplied a parameter that is not used by this function, or a duplicate parameter.

User response: Remove the parameter from the File Manager invocation or the control statement.

FMNBA117 Invalid value for parameter *parameter*

Explanation: You specified an incorrect value for a parameter.

User response: Correct the File Manager invocation or the control statement.

FMNBA118 Value of parameter *parameter* too long

Explanation: You specified an incorrect value for a parameter.

User response: Correct the File Manager invocation or the control statement.

FMNBA119 Value out of range for parameter *parameter*

Explanation: You specified an incorrect value for a parameter.

User response: Correct the File Manager invocation or the control statement.

FMNBA120 Invalid or inconsistent KEYLOC or KEYLEN value

Explanation: The key length value and key location value that you specified would result in a key that does not fit into the record where it is to be used.

The key location, plus the key length, minus 1 must be less than or equal to the record length. (For example, if the key location is 50 and the key length is 10, the record length must be 59 or more.)

User response: Change the key length, the key location, or both.

FMNBA121 Increment value too high

Explanation: The increment value does not fit within the key length specified.

User response: Change increment or key length.

FMNBA130 No print feature on SYSPUNCH device

Explanation: You called a function that interprets punched cards, but your card punch cannot interpret cards.

User response: Copy to cards without interpreting, or allocate SYSPUNCH to a card punch that has a print feature.

FMNBA133 *ddname* DD is allocated to a dummy device

Explanation: There is no data set available for processing. File Manager cannot process dummy data sets.

User response: Allocate a real data set, and rerun the function.

FMNBA134 FMNIIPRT DD must be same unit as SYSPUNCH DD

Explanation: The function you tried to use requires the FMNIIPRT DD to be allocated to the same device as SYSPUNCH.

User response: Allocate FMNIIPRT with unit affinity to SYSPUNCH.

FMNBA150 End of *xxxxx* tape sensed

Explanation: The indicated tape is positioned at the end of the tape (that is, after the end-of-tape (EOT) mark). If the tape is a reel, it might be pulled off the feeding reel.

User response: If the tape stops before its physical end, you can proceed with caution if required (for example to copy a broken tape to its physical end).

FMNBA156 End of *xxxxx* tape sensed, function terminated

Explanation: The indicated tape is positioned at the end of the tape (that is, after the end-of-tape (EOT) mark). If the tape is a reel, it might be pulled off the feeding reel by subsequent forward processing of the tape.

FMNBA160 Unit *unit* does not support Erase Tape

Explanation: You called the Erase Tape function, but the tape unit cannot perform the Erase Tape function.

User response: Mount the tape on an IBM 3400 tape unit (or its equivalent) to perform this function.

FMNBA162 Error on *xxxxx* tape, ECB *ecb*, CSW *csw*, sense 0-6 *sense*

Explanation: An error occurred on the indicated tape unit. If a read data check occurred, message FMNBA163 might follow.

User response: Save the message text. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

FMNBA163 Bypass record - B, ignore error - I, user correct - C

Explanation: An error was found reading from an input tape.

User response: Reply `B` to skip the record and read the next record. Reply `I` to use the record as it was read into the input buffer. Reply `C` to correct the record.

FMNBA170 No EOD delimiter is set

Explanation: You specified EOD as the number of files in a tape function, but you have not specified an EOD value with the SET function. There is no default EOD value for tapes.

User response: Specify a different number of files or use the SET function to define an EOD delimiter.

FMNBA171 Input record exceeds *nnnnn* byte buffer

Explanation: File Manager could not allocate a buffer large enough for the record.

User response: Rerun the function with more virtual storage.

FMNBA172 *nnnnn* byte input block exceeds *nnnnn* byte buffer

Explanation: File Manager could not allocate a buffer large enough for the block.

User response: Rerun the function with more virtual storage.

FMNBA181 No more data found on input tape

Explanation: The end of data on an input tape has been reached. (If a data check occurs immediately after a tape mark, File Manager assumes that the end-of-data has been reached.)

FMNBA191 Invalid element

Explanation: The File Manager XML parser has detected one of the following:

- Invalid XML where it was expecting to find an element definition.

- An element definition that isn't valid for the current XML schema.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA192 Invalid attribute for element <&ELEM>

Explanation: The File Manager XML parser has detected one of the following:

- Invalid XML where it was expecting to find an attribute definition.
- An attribute definition that isn't valid for the current XML schema.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA193 Duplicate attribute &ATTR specified for element &ELEM

Explanation: The attribute &ATTR has been specified more than once for element &ELEM at the offset &nn.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA194 End tag missing for element &ELEM

Explanation: Invalid XML - the expected end tag was not specified for element &ELEM.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA195 End tag invalid for element &ELEM

Explanation: Invalid XML - an incorrect endtag for the specified element was found.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA196 Invalid data beyond final end tag

Explanation: Invalid XML - data found beyond final end tag.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Either the length of the XML or XML sent is in error. Correct the XML input and retry.

FMNBA197 Severe internal error processing XML

Explanation: This message ise preceded by other messages that indicate what the severe error was. This is most likely to be an insufficient memory issue.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Analyze the preceding messages and action appropriately. You may need to contact your IBM support center.

FMNBA198 Data around error: '&DATA'

Explanation: This is an informational message to display the XML where the error was detected. Its is followed by FMNBA199 which has a vertical bar to the current position the parser was processing when the error occurred.

User response: None.

Programmer response: This message is followed by FMNBA199 which indicates where the error occurred. Analyze the preceding messages and action appropriately. You may need to contact your IBM support center.

FMNBA199 Error occurred: '&DATA'

Explanation: This informational message follows the FMNBA198 message to indicate the exact location of the error.

User response: None.

Programmer response: This message is preceded by FMNBA198, which displays data around where the error occurred. Analyze the preceding messages and action appropriately. You may need to contact your IBM support center.

FMNBA218 No disk record found

Explanation: The specified disk record does not exist on this track.

User response: If appropriate, specify a lower disk record number.

FMNBA219 No home address record

Explanation: A home address record was not found at the specified disk location. This is probably a hardware error.

User response: Rerun the function. If the problem recurs, contact your system support.

FMNBA220 Should new EOF be written after this record? Y or N

Explanation: The DRL function lets you change an end-of-file record into a record that has a KEY and DATA field (convert EOF to a data record).

User response: If you want a new end-of-file record to follow this record, reply `Y`. Otherwise, enter `N` or `U` to exit the function.

FMNBA230 Invalid element <&ELEM.> value: &VAL

Explanation: Either an attribute value or the data value for the specified element are invalid.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. It may be preceded by messages giving more detail as to the specific problem. Correct the XML input and retry.

FMNBA231 Invalid CDATA ']]>' missing <&ELEM.>

Explanation: A CDATA tag has been provided but no end CDATA string was found.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA232 Child element <&ELEM.> with wrong parent

Explanation: The element specified is a child element, but has been specified with the wrong parent element.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA233 <&ELEM.> not specified

Explanation: A required element has not been specified.

User response: None.

Programmer response: This message is followed by FMNBA665 or FMNBA666 and FMNBA667, which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA234 Library number &no exceeds number of <library> elements

Explanation: A `<member lib="&no" ...>` value has been specified where the library number exceeds the number of <library>data set name </library> value provided previously.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA235 Attribute &ATTR invalid value &VAL

Explanation: The attribute value specified is invalid for the current element.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA236 Template could not be updated - criteria would be lost

Explanation: During template editing an update was requested by providing the <copybooks> ... </copybooks> tags. The copybooks provided deleted fields that were referenced previously in criteria in the template and the template cannot be updated without this criteria information being lost.

User response: None.

Programmer response: Change the template to remove the field references that have been deleted in the latest version of the copybooks or change the copybooks to include the required field references and retry the update. Alternatively you can specify the REPLACE=YES option to override the existing template.

FMNBA237 id attribute not specified or invalid symbol number - &ID

Explanation: The <layout> id attribute is either missing or the value specified is not a valid 01 symbol reference number in the template.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA238 related01="&ID." attribute specified invalid symbol number

Explanation: The related01 attribute specifies a value that is not a valid 01 symbol reference number in the template for related ID. A related 01 must be a symbol reference of an 01 layout other than the current layout.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA239 type attribute not specified or invalid - &TYPE

Explanation: The <criteria > tag has been specified without the required type attribute or the type attribute does not specify one of the following values: ID, RID or SEL.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA240 byfield="1" required for dynamic template

Explanation: The <criteria > tag must specify byfield="1" attribute value for a dynamic template.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA241 Only type="SEL" allowed for dynamic template.

Explanation: The `<criteria>` tag must specify `type="SEL"`. Any other type value is invalid.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA242 Expression exceeds allowable length.

Explanation: The specified expression causes the template segment block to exceed the maximum allowed 32760 bytes.

User response: None.

Programmer response: Reduce the expression length. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA243 Expression error around offset &nn.

Explanation: The specified expression is invalid. The offset provided is the byte location into the expression where the error was detected.

User response: None.

Programmer response: Check any preceding error message and correct the expression. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA244 Symbol ref=&REF is invalid for current layout - Should be in range &LOW to &HIGH.

Explanation: The reference number specified on the <symbol> tag is invalid for the current layout. Must be a number in the range shown in the message.

User response: None.

Programmer response: Correct the reference number. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA245 Symbol ref=&ref out of order

Explanation: When you are creating a dynamic template the symbols references must be provided in sequence starting from 2.

User response: None.

Programmer response: Correct the reference number. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA246 Symbol ref=&ref start valure required

Explanation: When you are creating a dynamic template the start attribute must be specified.

User response: None.

Programmer response: Specify the start value. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA247 Symbol ref=&ref start value cannot be zero

Explanation: When you are creating a dynamic template the start attribute value cannot be zero for non offset start values.

User response: None.

Programmer response: Provide a positive integer value for start or specify offset="1" on the <symbol > tag. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA248 Symbol ref=&ref Invalid type attribute.

Explanation: When you are creating a dynamic template the type attribute must be one of these values:

C Alphanumeric
AN Alphanumeric
AX Alphanumeric displayed in long hexadecimal
VC Varying character
ZC Varying character null terminated
B Binary
BI Binary
P Packed decimal
PD Packed decimal
ZD Zoned decimal
FE External floating point
FP Internal floating point
BT Bit string
VB Varying bit
G Graphic string
VG Varying graphic

User response: None.

Programmer response: Specify a valid type attribute. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA249 Symbol ref=&ref Invalid length attribute.

Explanation: When you are creating a dynamic template the length attribute must be one of these values:

PD Must be between 1 and 16.
BI Must be 1, 2, 4 or 8.
ZD Must be between 1 and 32.
FP Must be 4 or 8.
FE Must be >6 and <24. The precision is the length minus 6. The scale is the precision minus 1.
DBCS strings Must be an even number of bytes.

User response: None.

Programmer response: Specify a valid length attribute. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA252 Output tape not positioned at load point or after EOF labels

Explanation: The output tape is not positioned correctly to write a labeled data set. Valid positions are the load point and after another labeled file.

User response: Ensure that the tape is positioned correctly and rerun the function, or rerun the function without label processing.

FMNBA253 No *xxxxxx* label found on input tape

Explanation: The tape indicated is not a standard label tape, or is not positioned correctly.

User response: Mount a labeled tape. Position it at the data set you want to process and rerun the function, or rerun the function without label processing.

FMNBA254 No *xxxxxx* label found on output tape

Explanation: The output continuation volume is not a standard label tape.

User response: Ensure that all output volumes are initialized with standard labels and rerun the function,

or rerun the function without label processing.

FMNBA260 Symbol ref=&ref heading too long.

Explanation: The <heading> value cannot exceed 20 bytes in length.

User response: None.

Programmer response: Specify a shorter heading. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA261 Symbol ref=&ref <createn> element invalid for non numeric symbol.

Explanation: The <createn> value should not be specified for the current symbol because its not numeric.

User response: None.

Programmer response: Remove the <createn> element. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA262 element value too long

Explanation: The value specified for the element exceeds the allowable length for this element.

User response: None.

Programmer response: Correct the length of the element data. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA263 Element in conflict with previously specified elements or type

Explanation: The element specified in the following FMNBA230 message cannot be specified here because it is dependant on specific attributes being set in a parent or it has exceeded the allowed occurrence number for this element. Here is a list of possible problems:

Template <scramble> specifications
- <translate> has been specified without `<scramble type="3">` being specified.
- <range> has been specified and conflicts with <scramble> type or a dsn attribute.
- <value> has been specified and conflicts with the <scramble> type or previous <range> specification.
- <sval> has been specified when a value list is not expected. Normally indicates a <scramble> dsn attribute was specified.

Editor <hex> specification
- <hex> tag has been specified more than once.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA264 Required attributes not specified.

Explanation: The element requires certain attributes to be specified as documented in the associated schema:
- <translate> has been specified without incol, outcol or dsn attributes.
- <range> has been specified without min or max attributes.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA265 Scramble translate attribute requires translate element.

Explanation: `<scramble type="3">` has been specified and a <translate> element is then expected to specify the translate options. The <translate> element has not been specified.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA267 <criteria> must specify byfield="1" for <byline> to be valid

Explanation: A <byline> element requires <criteria byfield="1"> and this has not been specified on the parent <criteria> element.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA268 <describe> exceeds maximum 5 elements.

Explanation: Too many <describe> elements have been specified.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA269 template required - command cannot be processed.

Explanation: A <describe> or <layout> element is being processed for a non-dynamic template and the TMPX session was invoked without a template.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA271 <copybooks> can only be specified for a copybook template

Explanation: You have specified the <copybooks> element for a non-copybook or dynamic template.

User response: None.

Programmer response: Remove the <copybooks> element and children. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665, or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA273 related01="&ID." attribute specified without type="RID" attribute

Explanation: A <criteria > element has been specified incorrectly. The `type="RID"` attribute must be specified if a related01 attribute has been specified.

User response: None.

Programmer response: Correct the XML to specify `type="RID"` attribute. This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA278 Element has exceeded maximum occurrences

Explanation: The element value as described in the following FMNBA230 message exceeds the maximum occurrences allowed for this element. See relevant schema.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA279 Required element or child element not specified.

Explanation: Either the element value or a required child element has not been specified. This is issued for the <replace> element when either <repfrom> or <repto> are missing, or when <repfrom> or <repto> element is specified without any data.

User response: None.

Programmer response: This message is followed by FMNBA230 to identify the element, and by messages FMNBA665 or FMNBA666 and FMNBA667 which display the line number and line text where the error occurred. Correct the XML input and retry.

FMNBA281 Error on *xxxxx* tape, not a valid Exported Stacked Volume

Explanation: The volume loaded for input to the EVC (copy) or EVL (list) function is not a valid Exported Stacked Volume. Either the volume has been corrupted or the wrong volume has been requested. Processing cannot continue.

User response: Correct the input and retry the request.

FMNBA282 Error on *xxxxx* tape, bad table of contents on input volume

Explanation: The volume loaded for input to the EVC (copy) or EVL (list) function has a corrupted Table of Contents. Processing cannot continue.

User response: Correct the input and retry the request.

FMNBA283 Error logical volume *xxxxx* not found in TOC of volume *vvvvvv*

Explanation: The volume loaded for input to the EVC (copy) or EVL (list) function does not contain the requested logical volume in its Table of Contents. Processing continues.

User response: Correct the input and retry the request.

FMNBA284 Error duplicate logical volume *vvvvvv* **in request list**

Explanation: A duplicate logical volume name has been entered into the logical volume request list. Processing does not continue.

User response: Correct the input and retry the request.

FMNBA298 Output volume *xxxxxx* **not long enough for logical volume** *vvvvvv*

Explanation: Output volume is not long enough to contain the logical volume being copied. This is a report line (not a message number) for the EVL list function.

User response: Mount an output volume of suitable length.

FMNBA299 Error code *nn* **processing Exported Stacked Volume**

Explanation: An unexpected error occurred while processing the input Exported Stacked Volume. The error code *nn* gives an indication of the problem:

01–03 Initial input volume positioning error. Possible drive error.

06–10 Error processing header labels on input volume. Possible bad input volume.

15–24 Error during Exported Stacked Volume verification process. This is not a volume created by the VTS export function, or the volume has become corrupted.

30–34 Error reading the Table of Contents file from the Exported Stacked Volume. The Table of Contents file on the Exported Stacked Volume may be corrupted.

40–48 Logical volume could not be correctly located on the Exported Stacked Volume. The volume may be corrupted.

49 Logical volume could not be correctly located on the Exported Stacked Volume. The drive on which the input Exported Stacked Volume is mounted may not support the locate block CCW, or the volume may be corrupted.

50–51 Error during copy process. Possible drive error.

55–58 Error during listing process. Possible drive error.

60–62 Error positioning output volume. Possible drive error.

66–79 Internal VTS record format error. The volume may be corrupted.

85–86 Error writing to output volume. Possible drive error.

90–93 Unable to obtain storage for work areas. Error codes 90–92 indicate that the storage has been requested above the 16M line. Error code 93 indicates that the storage has been requested below the 16M line.

95 General copy error. See earlier message for more detailed error code.

User response: Take corrective action according to the error code.

FMNBA301 *unit* **is invalid device for this function**

Explanation: The specified device has a type which cannot be used in the function you invoked. (For example, the input device for a disk function is a tape unit.)

User response: Use a valid device for this function, or use a File Manager function appropriate for the device specified.

FMNBA310 Data set *dsname* **not found**

Explanation: No data set *dsname* was found on the unit specified, or in the VSAM catalog used.

User response: Correct the data set name in the DD statement, in the File Manager invocation, or in the control statement.

FMNBA311 No *ddname* **DD statement supplied**

Explanation: *ddname* is a ddname referred by an INPUT or OUTPUT parameter, or any other ddname used by File Manager. Either the corresponding DD statement is missing, or the ddname was used and freed by a TLT function.

User response: Correct the JCL or the INPUT or OUTPUT parameter, and rerun the job. To access a tape again after a TLT function, end the job step and continue with another step.

FMNBA314 WARNING: *dsname* **in use.** *function* **continues without exclusive control**

Explanation: You invoked a disk update function for a data set already in use by another task.

User response: Ensure that your update request does not expose the data or the other task and continue, or run the function when the data set is available for exclusive control.

FMNBA315 Data set request denied by SVC99 validation routine

Explanation: A dynamic allocation request was rejected by your system's SVC99 validation routine. This is probably due to a restriction imposed by your system.

User response: Contact your system support.

FMNBA316 *ddname* **DD open failed, ABEND code** *abend-reason*

Explanation: The data set or tape specified by *ddname* could not be opened.

User response: Refer to the accompanying system message, and check the status of the data set or tape.

FMNBA317 Invalid *xxxxxx* **data set for function** *xxx*

Explanation: The function that you specified cannot be used with the data set that you specified. For example, a VSAM function cannot process a sequential data set.

User response: Specify a function capable of processing the data. You can use basic disk functions to inspect data with unknown organization.

FMNBA318 *type* **data set** *ddname macro* **failed, ABEND code** *xxxx-xxx*

Explanation: The indicated operation failed for the indicated data set.

User response: Check for any accompanying system messages, and refer to the appropriate system manual.

FMNBA319 *type* **data set** *ddname macro* **RC** *xxxx-xxxx*

Explanation: The indicated operation failed for the indicated data set.

User response: Check for any accompanying system messages, and refer to the appropriate system manual.

FMNBA320 No SYSPRINT DD statement supplied

Explanation: File Manager requires SYSPRINT to be allocated.

User response: Ensure that a SYSPRINT DD statement is supplied, then rerun the job.

FMNBA321 EOV for *xxxx* **failed, ABEND code** *xxxx-xxxx*

Explanation: An ABEND occurred during EOV processing for the indicated tape.

User response: Check for any accompanying system messages, and refer to the appropriate system manual. If the problem persists, contact your system support.

FMNBA322 SL and NON SL functions cannot be used in the same execution

Explanation: The first function that was used on the tape volume required standard label processing, but the function you are trying to use requires non-standard label processing. The tape volume must be reallocated for non-standard label processing.

User response: Allocate the tape for non-standard label processing and retry the function.

FMNBA323 SL and NEW requested for NON SL function

Explanation: The tape volume was allocated for standard label output processing, but the function you tried to use requires non-standard label processing.

User response: Allocate the tape for non-standard label processing and retry the function.

FMNBA331 Permanent I/O error on *input*|*output* **ECB** *hhhhhhhh*

Explanation: An unexpected return code was returned for a read or write operation (where *hhhhhhhh* is the ECB value for the associated operation).

User response: Refer to the corresponding system message for more information.

FMNBA332 Permanent data check on *input*|*output*, **CSW** *csw*, **sense** *code*

Explanation: This is probably a hardware error.

User response: Refer to the hardware device reference manual for an interpretation of the CSW and the sense bytes.

FMNBA333 Read error, incorrect block length on input/output

Explanation: A block of the input/output data set does not correspond to the data set's format description from the VTOC or DD statement.

User response: Change the DCB parameter of the DD and rerun the job, or try using a tape or disk function instead of a QSAM function to process this data set.

FMNBA350 Block size (*xxxx***) invalid for FIXED,** *recsize* **output**

Explanation: The block size of a fixed unblocked data set must be the same as the record size.

User response: Specify a different block size or a different record size.

FMNBA351 Input data length (*length*) not multiple of recsize (*recsize*)

Explanation: An input block or record has a length that is not equal to, or a multiple of, the output record size.

User response: Correct the input specification or use the SET PAD processing option to adjust the input records.

FMNBA352 Block size (*nnnnn*) exceeds maximum (*max*)

Explanation: For a QSAM output function, you specified a block size greater than the maximum supported value.

User response: Specify a smaller block size.

FMNBA353 Record size (*recsize*) exceeds maximum (*max*)

Explanation: A record exceeds the maximum record size allowed for the output data set. This message might be issued if an output data set has inconsistent RECSIZE and BLKSIZE parameters. For example:
- RECFMOUT=VB and BLKSIZE is less than RECSIZE + 8
- RECFMOUT=V, B, or DB, and BLKSIZE is less than RECSIZE + 4

User response: Specify a smaller record size, or change the output data set definition.

FMNBA354 Block size (*blksize*) not multiple of record size (*recsize*)

Explanation: A block consists of one or more records. The block size must be the same as the record size, or a multiple of the record size.

User response: Specify a different block size or a different record size.

FMNBA355 Record size (*nnnnn*) invalid for FIXED,*nnnnn* output

Explanation: The record size encountered is not compatible with the output specifications.

User response: Correct the input or output specification or use the SET PAD processing option to adjust the input records.

FMNBA356 Inconsistent record size (*nnnnn*) for FIXED,*nnnnn* input

Explanation: The record size encountered is not compatible with the input specifications.

User response: Correct the input or output specification.

FMNBA358 Inconsistent record length field X'*nnnnn*'

Explanation: While reading variable-length (blocked) input, File Manager found a data block whose record-length field contains either zero or a number greater than the physical length of the data block.

User response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

FMNBA359 REXX variable *var* is empty

Explanation: The indicated REXX variable does not contain a string, or is not defined.

User response: Initialize the REXX variable in your procedure with the desired value before invoking File Manager.

FMNBA360 REXX Stem count *var* invalid

Explanation: The indicated REXX variable does not contain a valid number, or is not defined.

User response: Initialize the REXX variable in your procedure with the correct stem count before invoking File Manager.

FMNBA361 Incorrect block length field X'*xxxx*', block *nnnn*, length *nnnn*

Explanation: The indicated block does not contain variable format data, or the value in the block descriptor field is different from the physical length of the data block.

User response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

FMNBA362 Incorrect record length field X'*xxxx*', block *nnnn*, offset *nnnn*

Explanation: The indicated block does not contain variable format data, or the value in a record descriptor field is invalid for deblocking.

User response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

FMNBA363 Incorrect spanned record segmentation, block *nnnn*, record *nnnn*

Explanation: The indicated block contains a segment of a variable spanned record which cannot be assembled to an entire record.

User response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

FMNBA370 Key positioning not possible

Explanation: You cannot specify a key position with control interval access or a NONINDEXED VSAM input.

User response: Remove the key position specification.

FMNBA371 Output data set not ESDS

Explanation: You are using the TV function with the *nfiles* parameter, to copy more than one tape data set to a single VSAM data set. The output VSAM data set must be an entry-sequenced data set.

User response: Specify an ESDS as the output data set.

FMNBA372 Inconsistent key length or key position

Explanation: You have copied records from one VSAM key-sequenced data set to another VSAM key-sequenced data set. The output data set has a different key position or key length than the input data set.

User response: Check that you really intended to change the key position or key length.

FMNBA375 VSAM *macro* RC *rc*, Error Code X'*xx*' *yyy*

Explanation: A VSAM macro returned with an unexpected code. The message includes the failing macro, the return and error codes, and, for some errors, an additional explanation. For more information, refer to the documentation of your current release of VSAM.

X'xx' is, in hexadecimal:
- The error byte of the ACB if OPEN or CLOSE failed
- The error byte of the RPL if POINT, GET, or PUT failed
- The contents of Register 0 if SHOWCB or TESTCB failed

yyy might present additional information about the error.

User response: Take corrective action as described in the documentation for your current release of VSAM.

FMNBA378 VSAM catalog RC *nn*, RSCD IGG0CL*xx* - *nnn*...

Explanation: Catalog management returned these return and reason codes as a result of a catalog error or exceptional condition. For an explanation of these codes, refer to the documentation of the system message IDC3009I. If the error is related to a single catalog entry, the defective entry is flagged in the SCS output and the function continues.

User response: Take the appropriate action based on the explanation of the codes.

FMNBA393 OPEN warning code X'*xx*' *yyy*. Continue ? Y or N

Explanation: VSAM OPEN returned a warning code indicated in the message. File Manager lets you decide whether to ignore the condition or to exit from the function. For more information, refer to the documentation of your current release of VSAM.

User response: Take corrective action as described in the documentation for your current release of VSAM. Enter `Y` to continue, or `N` to exit from the function.

FMNBA394 OPEN warning code X'*nn*'(condition). Processing continues.

Explanation: VSAM OPEN returned a warning condition. The reason code is documented in manual DFSMS Macro Instructions for Data Sets. File Manager attempts to continue processing.

Condition is a terse form of the explanation for some reason codes, as given by DFSMS Macro Instructions for Data Sets as follows:

```
X'4C' previously interrupted
X'60' data set flagged unusable
X'64' Alternate index not built
X'68' time stamp mismatch
X'6C' time stamp mismatch
X'74' not properly closed
```

User response: Consider the terse description and consult the DFSMS Macro Instructions for Data Sets manual for the OPEN reason code listed to determine if this condition is expected.

FMNBA395 Recovery from invalid RBA (*nnnnn*) failed

Explanation: VSAM rejected the RBA you entered. An attempt to locate the first record with a higher RBA also failed.

User response: Enter an RBA matching the beginning of a record.

FMNBA396 SHOWCAT for *input*|*output* data set failed, RC *rc*

Explanation: A VSAM SHOWCAT macro failed (giving return code *rc*) while searching for information about the specified data set. For an explanation of these codes, refer to the documentation for your current release of VSAM. If the SHOWCAT macro failed for a VSAM alternate index operation, it might show incorrect return codes because of VSAM recovery action.

User response: Take the appropriate action based on the explanation of the codes.

FMNBA397 DIV *req* request error, RC *nn*, Reason Code *rsn*

Explanation: A data-in-virtual macro has failed. The message includes the type of macro request and the reason code. For an explanation of the return code and reason code, refer to the documentation for your current release of your z/OS system.

User response: Take the appropriate action based on the explanation of the codes.

FMNBA401 *Input*|*Output* tape on *unit* is *vol1*, not *volser*.

Explanation: The tape that the operator mounted is not the volume requested.

User response: Ensure that you are using the correct volumes, check for incorrectly specified volume serial numbers, and rerun the function.

FMNBA407 Tape volume to be labeled on *unit*. Enter '*volser*', Cancel - U

Explanation: Standard label output was specified, but the named tape volume is not labeled. The system waits for a reply.

User response: Enter a serial identification to be used for the volume, or enter `U` to cancel.

FMNBA409 Unexpired file *name date* on tape *unit*,*vol1*. Ignore - I, Cancel - U

Explanation: The expiration date on the volume mounted as the output tape was not reached. The system waits for a reply.

User response: Ensure that the tape can be used for output and enter `I` to continue, or enter `U` to cancel.

FMNBA411 Ready *unit*, then press ENTER

Explanation: The unit *unit* is not ready. The system waits for a reply.

User response: Ensure that the correct device is attached and assigned. Ready the device and press Enter. Enter `QUIT`, `CANCEL`, `EXIT`, or `EOJ` to terminate.

FMNBA413 Manual unload/rewind detected on *unit*, press ENTER to rewind and continue

Explanation: A manual operator intervention occurred on the indicated unit. The system waits for a reply.

User response: Press Enter to rewind and continue, or enter `QUIT`, `CANCEL`, `EXIT`, or `EOJ` to terminate.

FMNBA420 Line &NO Syntax error detected at column &COL in the following statement.

Explanation: You are running a procedure with *FASTREXX ON specified and the procedure has failed FASTREXX syntax checking for the statement beginning at the line number specified. For continued statements the line number represents the first line for the statement. The column position is the position into the continued statement as printed. This message will be preceded by more detailed messages.

User response: None.

Programmer response: Check the preceding messages and correct the procedure. If this message is produced and a template expression is being run then an internal error has occurred and you need to contact your IBM Support Center.

FMNBA421 Keyword being processed: &KWD

Explanation: You are running a procedure with *FASTREXX ON specified and the procedure has failed FASTREXX syntax checking when processing the keyword referenced.

User response: None.

Programmer response: Check the preceding messages and correct the procedure.

FMNBA422 &statement

Explanation: This message is used to print the procedure statement (including continuations) that was being processed when the syntax error occurred, or to list the expected keywords when a keyword is processed that is not expected.

User response: None.

Programmer response: Check the preceding messages and correct the procedure.

FMNBA423 Error - one of the following keyword expected.

Explanation: This message indicates the keyword being processed is not expected and is followed by a list of expected keywords.

User response: None.

Programmer response: Check the following messages and correct the procedure.

FMNBA424 The function is incomplete - no closing parenthesis found

Explanation: A FASTREXX function has been coded incorrectly - a right parenthesis is missing.

User response: None.

Programmer response: Provide the missing parenthesis and rerun.

FMNBA425 The condition is incomplete

Explanation: A FASTREXX condition is incomplete - either the left hand side is expected or the right hand side has not been specified.

User response: None.

Programmer response: Correct the condition and rerun.

FMNBA426 Incomplete DO levels in procedure - END statements required

Explanation: A DO or SELECT statement does not have a matching END statement.

User response: None.

Programmer response: Provide the missing end statements and rerun.

FMNBA427 Incomplete quoted string - closing quote/apostrophe required

Explanation: A literal value has been coded that begins with a quote or apostrophe and no ending quote or apostrophe has been coded.

User response: None.

Programmer response: Provide the closing quote or apostrophe and rerun.

FMNBA428 Incomplete comment - closing */ required

Explanation: A REXX comment is incomplete.

User response: None.

Programmer response: Provide the closing */ and rerun.

FMNBA429 Invalid continuation comma on last statement in procedure

Explanation: A comma has been coded as the last non blank character in the procedure.

User response: None.

Programmer response: Correct by either removing the comma or providing the continued statement and rerun.

FMNBA430 Right parenthesis missing

Explanation: A conditional statement has unbalanced left and right parenthesis.

User response: None.

Programmer response: Provide the same number of right and left parenthesis in the conditional statement referenced and rerun.

FMNBA431 Invalid operator found in condition

Explanation: The specified operator is not an operator that has is supported for FASTREXX processing.

User response: None.

Programmer response: Correct the operator to be valid REXX operator and rerun.

FMNBA432 Keyword not supported for FASTREXX processing

Explanation: The keyword is either a REXX keyword or external function that is not supported for FASTREXX processing.

User response: None.

Programmer response: Either change the keyword to an equivalent FASTREXX supported keyword or remove the *FASTREXX ON card to allow REXX to process.

FMNBA433 Internal error - Format template ptr for &nn.th record type is zero.

Explanation: The runtime control blocks for a Template have been corrupted and an expression that relies on them cannot be run.

User response: None.

Programmer response: Contact your IBM Support Center.

FMNBA434 Internal error - FASTREXX Function &FUN not found in File Manager REXX function load module FMNRXFUN

Explanation: FMNRXFUN load module is not the same level or version as the FMNMAIN load module.

User response: None.

Programmer response: Installation problem. Check where these load modules are being run from and ensure your job is loading the correct modules. Contact your systems programmer and check your installation.

FMNBA435 Internal error - File Manager load modules FMNMAIN and FMNRXFUN are not at the same version/level

Explanation: FMNRXFUN load module is not the same level or version as the FMNMAIN load module.

User response: None.

Programmer response: Installation problem. Check where these load modules are being run from and

ensure your job is loading the correct modules. Contact your systems programmer and check your installation.

FMNBA436 Function &FUN function call failed rc = &RC.

Explanation: The specified function failed with the specified return code. This is normally a problem with the parameters passed to the function.

User response: None.

Programmer response: Check the preceding messages which should indicate the problem with the function parameters - correct and rerun.

FMNBA437 SIGNAL statement has no label

Explanation: A SIGNAL statement has been coded without a label value.

User response: None.

Programmer response: Provide the correct label value for the statement and rerun.

FMNBA438 Internal function coded incorrectly.

Explanation: This is an internal error that occurs running with a template. The internal function generated for the template expression is incorrect.

User response: None.

Programmer response: Contact your IBM Support Center.

FMNBA440 Line &nn Expression always true at column &col in the following statement

Explanation: Informational message to say the indicated condition is always true.

User response: None.

Programmer response: Correct the logic if required.

FMNBA441 Line &nn Expression always false at column &col in the following statement

Explanation: Informational message to say the indicated condition is always false.

User response: None.

Programmer response: Correct the logic if required.

FMNBA442 logical value not 0 or 1

Explanation: You have specified a constant for a condition and only a 0 or 1 maybe specified..

User response: None.

Programmer response: Correct the condition and rerun.

FMNBA443 Field reference

Explanation: This message may be produced when listing the expected keywords for message FMNBA423. It indicates that a field reference (#nn) is an allowed keyword.

User response: None.

Programmer response: Examine all related messages and correct the procedure accordingly.

FMNBA444 SIGNAL &label statement unresolvable

Explanation: A signal statement has referenced a label name that has not been coded in the procedure.

User response: None.

Programmer response: Either provide the label or correct the SIGNAL statement to refer to a valid label name and rerun.

FMNBA445 No executable logic in procedure

Explanation: The procedure is made up of non executable instructions and therefore cannot be run.

User response: None.

Programmer response: A procedure must have at least one statement that can be run to be a valid procedure.

FMNBA446 SIGNAL &lab causes infinite loop

Explanation: The SIGNAL &label causes an infinite loop because the specified label is the label for the SIGNAL instruction.

User response: None.

Programmer response: Correct the logic flow and rerun.

FMNBA448 Constant - numeric or character

Explanation: This message may be produced when listing the expected keywords for message FMNBA423. It indicates that a numeric or character literal is an allowed keyword.

User response: None.

Programmer response: Examine all related messages and correct the procedure accordingly.

FMNBA449 File Manager external function

Explanation: This message may be produced when listing the expected keywords for message FMNBA423. It indicates that a File Manager external function is an allowed keyword.

User response: None.

Programmer response: Examine all related messages

and correct the procedure accordingly.

FMNBA450 External CICS interface program DFHXCIS cannot be loaded

Explanation: DFHXCIS module cannot be loaded via an OS load service. Generally this means the cicshlq.SDFHEXCI has not been included in the tasklib, steplib, or link list, for the current File Manager function.

User response: None.

Programmer response: Check the setup and define the cicshlq.SDFHEXCI load library to the STEPLIB, or link list for a TSO or batch session, or FMNLIB or link list for the Problem Determination Tools Common Component Server, when using the File Manager Eclipse plugin. See the *File Manager for z/OS Customization Guide*.

FMNBA451 External CICS interface &FUNC failed Resp=&RESP, Reason=&REAS

Explanation: File Manager has attempted to access a CICS region using the external CICS interface and has failed to perform the function &FUNC with the response code and reason codes shown in the message. This is normally indicative of a setup problem

User response: None.

Programmer response: Ensure the steps documented in the File Manager Customization Guide section Setting up CICS access for File Manager Base and Eclipse plugin have been performed. If they have been completed successfully then refer to the manual CICS Transaction Server for z/OS CICS External Interface Guide and search for the function documented in the message. The response and reason codes are documented. Check them and action accordingly or contact your IBM Support Center.

FMNBA453 External CICS interface - DPL CICS LINK failed RESP=&RESP, RESP2=&RESP2, Abend=&ABCODE

Explanation: File Manager has attempted to access a CICS region using the external CICS interface and has failed to perform the a DPL CICS Link to the FMN3CICS program with the responses documented in the message. This would normally be a setup problem.

User response: None.

Programmer response: Ensure the steps documented in the File Manager Customization Guide section Setting up CICS access for File Manager Base and Eclipse plugin have been performed. If they have been completed successfully then refer to the manual CICS Transaction Server for z/OS CICS External Interface Guide and search for the function documented in the message. The response and reason codes are documented. Check them and action accordingly or contact your IBM Support Center.

FMNBA454 Failed to start server CICS applid &CICS failed RC=&RC RESP=&RESP, RESP2=&RESP2

Explanation: File Manager has attempted been unable to start a server task in the CICS region requested.

User response: None.

Programmer response: This message precedes FMNBA462 messages which can be used to determine why the server task failed. Contact your IBM Support Center for more assistance.

FMNBA455 Socket create failed Rc=&RC,Reason= &RSN

Explanation: A UNIX System Service BPX1SOC to create a TCPIP socket has failed with the reason codes specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA456 Socket connect failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1CON to connect a socket to the CICS region server task has failed with the reason codes specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA457 Send to CICS applid &CICS failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1SND has failed to send data to the CICS region for the reason code specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA458 Receive from CICS applid &CICS failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1RCV has failed to receive data to the CICS region for the reason code specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA459 Send failed for logoff command CICS applid &CICS failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1SND has failed to send the logoff command to the CICS region for the reason code specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA460 Socket close failed for CICS applid &CICS failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1CLO has failed to close a socket that had been established with the CICS region for the reason code specified.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual then contact your IBM Support Center.

FMNBA461 CICS server task failed to obtain port for CICS applid &CICS

Explanation: File Manager has tried to initiate the FM transaction as a background server task in the specified CICS system and the started task has either failed to respond within 25 seconds or has not been able to obtain a TCPIP PORT number to be used for the communications that follow.

User response: None.

Programmer response: Check the status of the CICS region to ensure a task can be started and ensure the CICS region has been setup for TCPIP communications. If the problem persists contact you IBM Support Center.

FMNBA462 &CICS_MESSAGE

Explanation: This message is preceded by FMNBA454 and contains the error message encountered by the server task during initialization processing.

User response: None.

Programmer response: Address the error described by the message text or contact you IBM Support Center.

FMNBA463 Socket getaddrinfo failed Rc=&RC,Reason=&RSN

Explanation: A UNIX System Service BPX1GAI has failed to get the address information using the PORT and host name provided by the CICS region. A connection to the CICS region cannot be performed.

User response: None.

Programmer response: Refer to the UNIX System Services Messages and Codes for the reason code in the message. If you cannot determine the problem from the manual, then contact your IBM Support Center.

FMNBA464 No FMNCICS DD to produce CICS Applid list

Explanation: A generic or blank CICS APPLID has been entered in the data set name field for a function that will attempt to produce a selection list of CICS APPLIDs from the FMNCICS DD and the FMNCICS DD has not been allocated to the session.

User response: None.

Programmer response: Perform the step to setup the FMNCICS DD as documented in the File Manager Customization Guide section Setting up CICS access for File Manager Base and Eclipse plugin.

FMNBA465 Invalid attributes for FMNCICS file must be fixed 80

Explanation: The data set identified by the FMNCICS DD statement does not have a logical record length of 80 or is not fixed blocked.

User response: None.

Programmer response: Perform the step to setup the FMNCICS DD as documented in the File Manager Customization Guide section Setting up CICS access for File Manager Base and Eclipse plugin.

FMNBA466 Member name required in FMNCICS DD

Explanation: FMNCICS DD refers to a PDS or PDSE and a member name has not been specified.

User response: None.

Programmer response: Specify the member name on the DD statement and rerun the function.

FMNBA467 The FMNCICS DD has no records - no applid list can be produced

Explanation: FMNCICS DD refers to an empty data set or member, so no generic selection list can be produced.

User response: None.

Programmer response: Edit the data set or member and provide a list of CICS applids and descriptions as specified in the File Manager for z/OS Customization Guide section Setting up CICS access for File Manager Base and Eclipse plugin.

FMNBA516 SSSARSN reason code &RC returned from IEFSSREQ Service request

Explanation: A nonzero reason code has been returned from the IEFSSREQ service function SSOBSSMS. The reason code is documented in the IBM-supplied IEFSSSA macro.

User response: Contact your systems programmer.

FMNBA517 Cannot connect to DB2, CAF *request* **RC** *xx*, **Reason Code** *xx*

Explanation: File Manager failed to connect to DB2. The indicated Call Attach Facility request did not complete successfully.

User response: Contact your DB2 administrator.

FMNBA518 Unexpected RC *xx* **returned from IEFSSREQ** *request*, **token** *nnnnnn*

Explanation: A subsystem request to retrieve OAM storage group information from SMS failed with an unexpected return code.

User response: Contact your system support.

FMNBA519 Function terminated by OAM environment error

Explanation: File Manager failed to establish connection with OAM.

User response: Contact system support.

FMNBA520 OAM *req* **failed, RC** *rc*, **Reason Code** *rsn*

Explanation: The requested OAM function failed.

User response: For an explanation of the return code and reason code, refer to the documentation for your current release of z/OS.

FMNBA521 Directory *request* **failed, DB2 not accessible**

Explanation: The requested OAM function failed. File Manager could not access DB2.

User response: Contact your DB2 administrator.

FMNBA522 Directory *request* **failed, DB2 privilege not granted**

Explanation: Your user ID is not authorized to access the object directory tables as required.

User response: Contact your DB2 administrator. For a list of tables, refer to the *File Manager Customization Guide*.

FMNBA523 Directory *request* **failed, unexpected SQL RC** *rc*

Explanation: The requested OAM function failed. An unexpected return code was received from SQL.

User response: Contact your DB2 administrator.

FMNBA524 OAM STORE failed, collection not found or definition denied

Explanation: An OSREQ STORE request failed because the target collection could not be found, or the user is not authorized to define a new one.

User response: Check your installation SMS specifications for correct class values if necessary.

FMNBA525 Directory *request* **failed with RC** *rc*, **trying again...**

Explanation: The requested OAM function failed with an SQL return code indicating a locked DB2 resource. This is an information message only. File Manager continues and reissues the SQL request.

FMNBA526 Directory *request* **failed with SQL RC** *rc*, **DB2 resource locked**

Explanation: The requested OAM function failed with an SQL return code indicating a locked DB2 resource.

User response: Rerun the function. If the problem persists, contact your DB2 administrator.

FMNBA529 Skip count exceeds *bytes* **byte object size**

Explanation: You asked to skip more bytes than the object contains. You have effectively asked File Manager not to print any data.

User response: Specify a lower skip count value.

FMNBA530 No object header found on input, specify collection and object name

Explanation: The input data did not contain object header records, thus no defaults for collection and object name are available.

User response: Specify the collection and object name and rerun the function. To create input data with

header records, use functions OS or OV with the HEADER option.

FMNBA534 STOR/MGMT class value overridden by defaults

Explanation: An OSREQ STORE has completed, but it could not use the Storage Class or Management Class that you specified. Default values for the collection were used instead.

User response: Check your installation SMS specifications for correct class values if necessary.

FMNBA538 Input data not in object dump format

Explanation: An input record was found with unexpected contents, or end of input occurred before the end of an object.

User response: Correct the input data and rerun the function.

FMNBA539 *nnn* object(s) could not be dumped

Explanation: Backup of one or more objects failed. File Manager issues this message at the end of the function. Message FMNBA200 shows the OAM error information for each failing object.

User response: Check the File Manager output for message FMNBA200 and take appropriate actions.

FMNBA590 Format of DB field incorrect at offset *nnnnn*

Explanation: The format of the indicated double-byte character set field is incorrect. File Manager found a double-byte field that has an odd length.

File Manager issues this message for only the first incorrect field it finds. It prints this message at the end of the record, and replaces the last character with an EBCDIC blank (X'40'). The function continues.

FMNBA591 Mixed field ends before end of DB subfield at offset *nnnnn*

Explanation: The double-byte character set (DBCS) subfield, at the offset shown, was not reset at the end of a mixed DBCS and EBCDIC field. For example, there might be no shift in (X'0F') character marking the end of a double-byte field in mixed DBCS and EBCDIC data.

File Manager issues this message for only the first incorrect field it finds. It prints this message at the end of the record. The function continues.

User response: Correct the field definition in the FMT command. Rerun the function.

FMNBA592 One or more records with incorrect DB fields found

Explanation: File Manager has detected incorrect double-byte character set fields in one or more records.

File Manager prints this message at the end of the listing. It indicates that File Manager issued one or more error messages. The function continues.

User response: Check the listing for message FMNBA590 or FMNBA591 to find the incorrect record.

FMNBA600 Minimum requirement to run this level of File Manager is z/OS

Explanation: To run File Manager Release 1, you need z/OS.

User response: You cannot run File Manager on your system.

FMNBA601 Minimum requirement to run File Manager is DFP 3.3

Explanation: To run File Manager for z/OS, you need MVS/DFP Version 3 Release 3 or later.

User response: You cannot run File Manager on your system.

FMNBA604 Load of module '*name*' failed, module not found

Explanation: File Manager cannot find a module which is required for your File Manager invocation. This module might have been deleted or protected (for example, if the functions supported by this module are not needed at your installation).

User response: If you need to use this function, contact your system support.

FMNBA605 Successfully recovered from ABEND

Explanation: An ABEND occurred while File Manager was running. Depending on the circumstances, a dump might be produced.

User response: Check for any system error or File Manager message and take corrective actions. If the problem persists, contact your system support.

FMNBA606 FILE MANAGER ENCOUNTERED AN INTERNAL ERROR - CODE *NNNN*

Explanation: File Manager encountered an internal problem. The error is identified by the File Manager error code.

User response: Contact your system support.

FMNBA607 FILE MANAGER ABENDED WITH CODE *CODE* **- REASON** *REASON*

Explanation: An abend occurred while File Manager was running. *code* is the z/OS abend code; *reason* contains additional information that is available for some cancel codes and abend codes.

User response: Refer to the documentation for your current release of the z/OS system.

FMNBA608 Load of module *xxxxxxxx* **failed, ABEND code** *nnnn-nnn*

Explanation: An attempt to load the specified module failed with an unexpected error. *nnnn-nnn* is the z/OS ABEND and REASON code.

User response: Refer to the accompanying system message or the system code, and take appropriate action.

FMNBA609 Access to REXX variable pool failed, *xxx* **R15** *nn*

Explanation: During an attempt to establish connection with the REXX variable pool system services failed.

User response: Contact your system support.

FMNBA610 Unexpected code returned from REXX variable interface, RC *nn*

Explanation: During an attempt to read from or write to the REXX variable pool the REXX system interface returned an unexpected result.

User response: Contact your system support.

FMNBA613 I/O error on *unit***, ECB** *ecb***, CSW** *csw***, sense 0-3** *sense*

Explanation: An unrecoverable I/O error occurred on the indicated device.

User response: Ensure that you are using the correct device, and that it is set online correctly. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

FMNBA620 XML input data set must be sequential.

Explanation: This is issued when the XML input is not on a sequential data set or not a member in PDS or PDSE. The data set organization is not supported.

User response: None.

Programmer response: Change the XML input so it is coming from a sequential data set or a member of a PDS or PDSE and rerun the job.

FMNBA621 name=&NAM attribute error. Invalid 01 name

Explanation: Either the <layout name='value'> or the <criteria rname='value'> have referred to a field name that cannot be matched to an 01 layout name in the current template.

User response: None.

Programmer response: Change the name or rname value to be a valid 01 name for the current template.

FMNBA622 copybook=&CPY attribute error. Name cannot be matched or previously processed

Explanation: The copybook attribute of the <layout> element refers to a 1 - 8 byte copybook member name. This member must be a valid copybook that was used to build the template. The <layout copybook='&CPY'> can only be repeated for the number of 01s in the referenced copybook. You have either specified a copybook name that isn't referenced in this template or you have repeated the definition more times than the number of 01 layouts referenced in this copybook.

User response: None.

Programmer response: If you are updating an old template (REPLACE=NO) and you haven't specified <copybooks> then you may need to rebuild this template so the copybook names can be used for the <layout>. Do this by specifying the relevant <copybook> elements. Otherwise correct the copybook name or remove the excessive <layout> elements and rerun the process.

FMNBA623 name=&NAM attribute error. Field name cannot be matched

Explanation: The name attribute value specifies a field name that cannot be located in the current template.

User response: None.

Programmer response: Correct the field name and rerun the process.

FMNBA628 A fully qualified member name is required

Explanation: You have specified either an output member mask or not specified an output member name when the input data set is sequential or you have specified an output sequential file without a blank or generic input member name.

User response: None.

Programmer response: Specify a valid member name and rerun the process.

FMNBA629 Template exported to XML successfully

Explanation: Successful completion of an export function.

User response: None.

Programmer response: None.

FMNBA630 &MEMCNT members read &UPCNT Exported &REPCNT Not replaced &ERRCNT Errors

Explanation: Statistical summary message for export template function

User response: None.

Programmer response: None.

FMNBA631 &MEMCNT members read &UPCNT Exported &REPCNT Export replaced &ERRCNT Errors

Explanation: Statistical summary message for export template function

User response: None.

Programmer response: None.

FMNBA634 &MEMCNT members read &UPCNT Imported &REPCNT Import Updated &ERRCNT Errors

Explanation: Statistical summary message for import template function

User response: None.

Programmer response: None.

FMNBA635 &MEMCNT members read &UPCNT Imported &REPCNT Import replaced &ERRCNT Errors

Explanation: Statistical summary message for import template function

User response: None.

Programmer response: None.

FMNBA649 SVC99 *type* failed, RC *rc*, ERR *ec*, SMS RSCD *ic ddname*

Explanation: The indicated SVC99 (allocation) operation failed. The failure occurred during SMS processing.

User response: For explanations of *rc, ec,* and *ic*, refer to the documentation for your current release of z/OS.

FMNBA650 SVC99 *type* failed, RC *rc*, ERR *ec*, INFO *ic ddname*

Explanation: The indicated SVC99 (allocation) operation failed.

User response: For explanations of *rc, ec,* and *ic*, refer to the documentation for your current release of z/OS.

FMNBA651 Locate macro failed, RC *rc*, REG0 X'*xxxxxxxx*'

Explanation: Unexpected return code from the LOCATE macro.

User response: For an explanation of the return code and register contents, refer to the documentation for your current release of z/OS.

FMNBA652 TIOT scan error, *ddname* not found

Explanation: An error occurred while scanning the z/OS control block TIOT. This is probably a File Manager error.

User response: Exit File Manager and rerun the function. If the problem persists, contact your system support.

FMNBA653 DSCB/JFCB read for *xxxxxxx* failed, RC *rc*

Explanation: Unexpected return code when reading the DSCB or JFCB.

User response: For an explanation of the return code, refer to the descriptions of the macros OBTAIN and RDJFCB in the documentation for your current release of z/OS.

FMNBA654 UCB scan error, RC *rc*, Reason Code *rsn*

Explanation: A UCB scan operation (to determine the device) failed with an unexpected return code.

User response: For an explanation of the return code, refer to the description of the UCBSCAN macro in the documentation of your current release of z/OS.

FMNBA655 TSO Service Facility call error - RC *rc*, Reason Code *rsn*

Explanation: An unexpected return code was received from the TSO Service Facility. The TSO command was not executed.

User response: For an explanation of the return code and reason code, refer to the *z/OS TSO/E Programming Services*.

FMNBA661 Template imported from XML successfully

Explanation: Successful completion of an import function.

User response: None.

Programmer response: None.

FMNBA665 XML error occurred on line &LINE at offset &OFFSET

Explanation: This message provides the line number and line offset where the XML error occurred.

User response: None.

Programmer response: Look at the previous messages to determine the type of error, correct the XML line referenced and rerun the process.

FMNBA666 XML error occurred on line &LINE

Explanation: This message provides the line number where the XML error occurred.

User response: None.

Programmer response: Look at the previous messages to determine the type of error, correct the XML line referenced and rerun the process.

FMNBA667 Line:&line value

Explanation: This message displays the line where the XML error occurred.

User response: None.

Programmer response: Look at the previous messages to determine the type of error ,correct the XML line referenced and rerun the process.

FMNBA668 Function must be run with File Manager for IMS to process IMS template, views or criteria sets.

Explanation: You cannot create or update an IMS template view or criteria set when running the import function under the base product. It must be run from the FM/IMS template options if running in full screen mode or using the program FMNIMS for batch processing.

User response: None.

Programmer response: Rerun the function in an FM/IMS environment.

FMNBA669 Function must be run with File Manager for DB2 to process DB2 templates..

Explanation: You cannot create or update a DB2 template when running the import function under the base product. It must be run from the FM//DB2 template options if running in full screen mode or using the program FMNDB2 for batch processing.

User response: None.

Programmer response: Rerun the function in an FM/DB2 environment.

FMNBA670 XML template type does not match the existing target template to be updated.

Explanation: You are trying to import a template from XML with REPLACE=NO meaning the XML is used to update the existing template. The existing template is as different type to the XML template being imported.

User response: None.

Programmer response: Either change the type on the <template type='value'> element to match the template to be updated or change the import to specify REPLACE=YES.

FMNBA671 Element <&ELEM.> exceeds the maximum occurrence of &NO.

Explanation: The XML element is repeated more than the allowed number of occurrences for that element.

User response: None.

Programmer response: Correct the XML to remove the unnecessary element definitions and rerun the process.

FMNBA672 Element <&ELEM.> invalid for &TYPE template.

Explanation: The element is not valid for the type of template being processed.

User response: None.

Programmer response: Remove the element definition or correct the <template type='value'> to reflect the correct template type for the element.

FMNBA673 Copybook <member> element has not been found for segment=&seg.

Explanation: You are trying to import a IMS template from XML and you have not provided copybook definitions to describe all the segments defined to the corresponding data base.

User response: None.

Programmer response: Provide <member name='name' segname='segment'> values for all segments in the referenced DBD.

FMNBA674 One of the IMS template libraries is not a PDS or PDS/E data set.

Explanation: The associated template data set for a view or criteria set import is not a PDS or PDSE data set. This could have been specified using the <imstp>data set name</imstp> element. The data set name may also be obtained from the existing view or criteria set for REPLACE=NO, or the existing template definitions for FM/IMS found either on the panels or in the options module definition.

User response: None.

Programmer response: Correct the IMS template library definitions and rerun the process.

FMNBA675 Template &TP for View or Criteria set was not found.

Explanation: The associated template member for a view or criteria set cannot be found in the current template libraries. The template member name is obtained from the <dbd>name</dbd> definition or for REPLACE=NO can be obtained from the existing view or criteria set definition.

User response: None.

Programmer response: Ensure the associated template member can be found. This may require providing <imstp>data set name</imstp> elements to identify the location of the template member.

FMNBA676 Template &TP for View or Criteria set is corrupted.

Explanation: The associated template member for a view or criteria set cannot be loaded because it contains invalid data for an FM/IMS template.

User response: None.

Programmer response: Recreate the template member or correct the template data sets (<imstp> definitions) and ensure the associated template member being loaded is a valid FM/IMS template.

FMNBA677 Cannot create a template of type: &type

Explanation: The import routine cannot create the template for one of the following reasons.

- The template is a BASE or IMS template and <copybook> elements have not been supplied to create the template.
- The template is a DB2 template and <db2object> element has not been provided.
- The template is an IMS view or criteria and <imstp> and <dbd> elements have not been provided.
- The template you are trying to import is an IMS extract template - this type of template can only be updated. REPLACE=NO must be specified and the template must exist.

User response: None.

Programmer response: Provide the <copybooks> elements so the template can be created.

FMNBA678 Attribute &ATTR invalid for &TYPE template.

Explanation: The attribute specified is not allowed for the type of template being processed.

User response: None.

Programmer response: Either remove the attribute or change the template type to a type that supports the attribute referenced in the message.

FMNBA679 <dbd> element is require for &TYPE template.

Explanation: You are trying to create a FM/IMS view or criteria set and you have not specified <dbd>name</dbd> which is required to create a view or criteria set.

User response: None.

Programmer response: Provide the associated DBD name via the <dbd>name</dbd> elements and rerun the process.

FMNBA719 Change failed

Explanation: The File Manager Find/Change utility could not change one or more strings because:

- The change string is longer than the find string and there is insufficient space in the record to accommodate the change string.
- An attempt is being made to change a VSAM KSDS key field, The key field will be identified with a 'K' next to the record number in the report.

User response: Change the find or change string to avoid the problem.

FMNBA820 Key area not mapped

Explanation: The layout specified does not map the key area of the record. Therefore a formatted key cannot be displayed.

User response: Either select a layout that can map the key field or position on a record mapped by a layout and enter FKEY with no layout.

FMNBA821 Layout not found

Explanation: The name you entered as a filter did not match any of the layouts in the current template.

User response: Use the "S" or "/" prefix commands to select a layout from the current listed layouts.

FMNBA822 No layout selected

Explanation: The user has exited the 01 Layout List Panel without selecting a layout. The FKEY command processing is terminated.

User response: None.

FMNBA823 Template required

Explanation: Formatted key processing requires a copybook or template to be provided to the edit, view or browse function.

User response: Specify the template or copybook that you require to map the key area.

FMNBA830 Picture string invalid

Explanation: You cannot use a picture string with numeric, bit, or unicode fields.

User response: This message is only produced if only numeric bit or unicode fields are the target of the search or change command. Either remove the picture clause, or change the scope so character fields can be searched.

FMNBA999 File Manager problem - message *nnnn* not found in table

Explanation: File Manager cannot find the message number *nnnn* in the message table. This is probably a File Manager error.

User response: Contact your system support.

FMNBB039 FASTPROC OUTFIL statement is not supported for this type of output data set

Explanation: You ran a DSC function containing FASTPROC OUTFIL statements, and one of the following is true:

- The input data set has fixed length records, the output data set has variable length or undefined format records, and one of the following conditions is true:
  - The output record length is undefined
  - There is a user procedure or a template being used
  - The data is packed
  - The output is a PDS(E)
- The input data set has variable length or undefined format records, the output data set has fixed length records, and one of the following conditions is true:
  - There is a user procedure or a template being used
  - The output data set has longer records
- The input data set is a QSAM data set with spanned records with length exceeding 32747 and the output is a VSAM data set
- the output data set is a VSAM VRDS data set and the user requested a MOD disposition (instead of OLD)

User response: Recode the DSC function to avoid using FASTPROC OUTFIL statements. For example, you might need to recode it to use REXX statements in a REXXPROC.

FMNBB172 Error accessing CICS resource func=*function* Cond=*condition* Resp2=*resp2*

Explanation: CICS returned a non-zero condition after attempting to perform the listed function against the selected CICS resource.

User response: Look up the listed *condition* and *resp2* values in the *CICS Application Programming Reference* or *CICS System Programming Reference* for further information about why the request failed.

In most cases (but not all) the listed function is the CICS command that returned the non-zero condition. Below is a list of possible values for function and the CICS command they relate to:

Function value	CICS command
INQSTART	INQUIRE
INQNEXT	INQUIRE
INQUIRE	INQUIRE
STARTBR	STARTBR
READ	READ
ENDBR	ENDBR
WRITE	WRITE
REWRITE	REWRITE
DELETE	DELETE
OPEN	SET (OPEN)
CLOSE	SET (CLOSE)
SET	SET
SYNCPT	SYNCPOINT
DELETEQ	DELETEQ
ENQ	ENQ
DEQ	DEQ
INQGEN	INQUIRE
SET TASK	SET (TASK)
READNEXT	READNEXT
READPREV	READPREV
READUPD	READ (UPDATE)
DELQ	DELETEQ
READQ TD	READQ (TD)

FMNBB331 Fields referenced in the TO template don't exist in the FROM template.

Explanation: The To template has existing mapping and at least one of the fields it has been previously mapped to cannot be found in the from template. This indicates that the TO template has been either mapped to a different from template, or the from template has changed so that fields previously mapped cannot be found.

User response: If you are running under ISPF you should be given an option to regenerate the mapping fields and continue. For batch jobs, DSM, and DSC functions, the keyword CORRESP=YES can be used to regenerate a corresponding mapping. This way you can rerun the job with that keyword specified to override the existing mapping.

FMNBB537 Not running from APF Authorized libraries. Check STEPLIB/LINKLIST.

Explanation: File Manager, after being launched from the PDTCC server, detected that it is not running APF authorized, which is required.

User response: Check that the steplib or linklist libraries have been correctly authorized. See also the *File Manager Customization Guide*.

FMNBB538 Terminating, not APF Authorized.

Explanation: File Manager is shutting down. This message is issued after FMNBB537.

User response: As for FMNBB537, check that the steplib or linklist libraries have been correctly authorized. See also the *File Manager Customization Guide*.

FMNBB631 You can only append to CICS files - &CMD cannot be performed.

Explanation: A CREATE, REPLACE, SAVEAS command has been attempted to a CICS file resource These commands are not supported.

User response: None.

Programmer response: None.

FMNBB632 JCL option is not supported for CICS resources.

Explanation: Option 3.6 or FCH batch with the JCL option is being run against a CICS resource. The option is not supported.

User response: None.

Programmer response: Rerun the function with the option unspecified.

FMNBC250 Unix service &serv failed

Explanation: A Unix system service was called, but returned a failure. Message FMNBC251 will follow. The &serv may be one of:

```
MVSSIGSETUP
SIGACTION
ACCEPT
READ
WRITE
WAIT
SOCKET
SETSOCKOPT
SET KEEPALIVE
BIND
LISTEN
ASYNCIO
SPAWN
CLOSE
PASSWORD
GETPWNAM
SETUID
EXIT
ATTACH
```

User response: Examine the following message (FMNBC251)

FMNBC251 Unix Return value &rv, Return code &rc, reason code &RSN

Explanation: After a Unix system service call failure, this lists the return value, code and reason code.

User response: Check the meaning of the reason code by either using BPXMTEXT (if it is available on your system, you can cut and paste the &RSN code and issue, from TSO, the command BPXMTEXT reasoncode, for an explanation of the Unix system service reason code) or by consulting the manual *UNIX System Services Messages and Codes*.

FMNBC515 FMNOPTS allocated to name.

Explanation: File Manager has detected that the FMNOPTS DD statement is allocated to name, where name is either the name of the data set allocated to FMNOPTS, or "SYSIN" indicating that the FMNOPTS statements have been provided instream. A listing of the options read from FMNOPTS follows.

User response: None.

FMNBC516 Error processing FMNOPTS options file.

Explanation: File Manager has encountered one or more errors while processing the options file allocated to the FMNOPTS DD statement. This message is preceded on one more error messages describing the error(s) encountered.

User response: Refer to the previous error messages related to the FMNOPTS options file. Correct the errors, and then rerun File Manager.

FMNBE052 Template data invalid for load Rsn=*nnn*

Explanation: The File Manager template is invalid. It may have been corrupted by editing it outside of File Manager.

User response: If this is not the case, then please report the problem to your Support Representative, stating the Reason Code.

FMNBE094 COBOL compile failed with Language Environment RC &RC and RSN &RSN

Explanation: When File Manager invokes COBOL compiler Version 5 (and above), it may terminate with a Language Environment® Abend Code which may be found in the Language Environment publication. If Abend code is 4093 and Reason code X'1C' or 28, then there is insufficient storage to run the compiler.

COBOL Version 5 requires at least 100MB of storage.

User response: The Language Environment abend and reason codes can be found in z/OS Language Environment Run-Time Messages manual.

FMNBE491 No DB2 TABLE matches the value entered. It is possible a DB2 object (eg view) that matches the name entered exists - however the selected DB2 utility supports DB2 tables ONLY.

Explanation: This message is issued when a DB2 object name - other than a table - is entered when preparing a DB2 LOAD or UNLOAD utility job. The DB2 LOAD and UNLOAD utilities support only table objects - not views, aliases, synonyms etc.

User response: Ensure that the DB2 object name specified is a DB2 table.

FMNBE719 Change failed

Explanation: The Find/Change utility has terminated with a nonzero return code.

User response: The explanation of the associated return code can be found in "FCH (Find/Change)" on page 1015.

FMNBE805 SMF logging error. Value returned by BPX1SMF: RC=&rc Reason Code=&RSN(Dec)

Explanation: File Manager is set up to write audit records to SMF for the current function. A failure occurred while attempting to write an audit record to SMF. The BPX1SMF call resulted in return code &rc and reason code &RSN, both expressed in decimal. The current function may be terminated.

User response: This error usually indicates a set-up problem. Possible reason for an auditing failure to SMF are:

1. The userid does have READ access to the SAF FACILITY class profile BPX.SMF.
2. The FMN0POPT module does not have a valid number specified for the SMFNO parameter.

Consult the systems programmer for assistance in resolving the cause of the error.

FMNBE808 CCSID not valid

Explanation: The specified CCSID is not known by this Z/OS system.

User response: Change the CCSID to one that is known by this Z/OS system.

FMNBE895 CCSID display off

Explanation: The CCSID column is not displayed in single formatting.

User response: None

FMNBE898 CCSID display on

Explanation: The CCSID column is displayed in single formatting.

User response: None

FMNBE991 Not selected record

Explanation: When the FMAP command has been entered the cursor is not positioned on a selected record or the record at the top of screen is not a selected record.

User response: Identify the record containing the field to be mapped by positioning the record at the top of the screen or position the cursor on the record when the FMAP command is entered.

FMNBE992 No field selected

Explanation: The field to be mapped has not been selected on the Field Selection List panel.

User response: Reissue the FMAP command and select a field from the list supplied in the Field Selection List panel.

FMNBE993 No template provided

Explanation: The template to be used to map the field has not been provided.

User response: Supply Template or Copybook details on the FMAP Copybook or Template panel

FMNBE994 Data may be truncated

Explanation: The template being used to map the field is larger than the field and data may be truncated.

User response: None

FMNBE998 No column selected

Explanation: The column to be mapped has not been selected on the Field Selection List panel.

User response: Enter the FMAP command and select

the column to be mapped on the Field Selection List panel.

FMNBE999 Field reference invalid

Explanation: The field reference specified in the FMAP command cannot be found in the current template.

User response: Specify the correct field reference in the FMAP command.

FMNBF003 Not enough space.

Explanation: The data is displayed in the CCSID of the terminal, but is stored internally in UNICODE. After conversion to UNICODE there are not enough characters to pad the field with UNICODE spaces.

User response: Remove one or more characters using HEX ON to see the characters in UNICODE.

FMNBF123 Converted UNICODE data too big for receiving field. Field *name* **length** *lllll***.**

Explanation: The receiving field *name* is not large enough to hold the converted UNICODE field of length *lllll*.

User response: Increase the size of the receiving field in the template by either changing the copybook and regenerating the template or editing the dynamic template and changing the field size.

FMNBF124 Return code *rc***, reason** *code* **rsn from CUNLCNV routine. Field** *name***.**

Explanation: The routine CNVLCNV could not convert the field *name*.

User response: For an explanation of the return code and reason code refer to the z/OS V1R11.0 Support for Unicode Using Unicode Services.

FMNBF427 Program Binder does not recognize member as a Load Module / Program Object. Function=&fc RC=&rc reason=&rsn.

Explanation: The File Manager View Load Module utility encountered an Binder API error when accessing the module member. The &fc is an internal code to represent the IEWBIND function being executed. The Binder API returned return code &rc and reason code &rsn.

User response: Refer to the manual SA22-7644-xx MVS™ Program Management: Advanced Facilities for an explanation of the Binder API return codes and reason codes.

FMNBF434 The module member does not contain any CSECT names.

Explanation: The File Manager View Load Module utility found no CSECT names in the module member and therefore cannot provide any information.

User response: None.

Appendix B. XML definitions for a template

Table 21. XML definitions for a template

Lvl	Element	Attribute	Type	Description
1	template	lang	BASE IMS	For templates derived from copybooks. Values: • COBOL • PLI • ASM
		type	All	Type of template. Values: • BASE • IMS • IMSVIEW • IMSCRIT • DB2 • DYNAMIC
		segmented	BASE	Applies to base templates only and is a Boolean value to indicate the template is segmented or not. Values: • 0 • 1 • true • false
2	describe		all	This is a template description. You can provide up to five occurrences of this element and it equates to five description lines. Maximum length is 54 characters per description line.
2	db2object		DB2	This is the DB2 object in the form owner name.objectname.
2	ssid		DB2	This is the four character DB2 subsystem. This value has no effect for import which uses the SSID of the DB2 session for which it is run.
2	db2rel		DB2	The DB2 release for the template object. This is for information purposes only.
2	dbd		IMS IMSVIEW IMSCRIT	This is the DBD name. It is used as the originating template name for view and extract criteria, when an import is being performed.
2	imstp		IMS IMSVIEW IMSCRIT	This is the IMS template data set name. This overrides the default template data set names provided in the options when importing a view or extract criteria.
2	dbdlib		IMS IMSVIEW IMSCRIT	This is the IMS DBDLIB that is used to build the IMS template. You can specify this element up to six times to provide more than one DBD library to be searched.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
2	relcrit	max	IMSCRIT	Relation criteria definitions. max="*nnn*" Integer Enter a number to be used as the default maximum number of target database records to be selected per source database record for the listed logical and application relationships. If you do not specify, there is no default maximum selections per database record.
		sel		Boolean Values: **1** Selected **0** Not selected
3	reldbd	name		name="*DBD name*" Source DBD name
		seg		seg="*segment name*" Source Segment
		targ		targ="*DBD name*" Target DBD name
		bidir		Boolean and appears on export (`bidir="1"`) if the relationship is bidirectional.
		sel		Boolean Values: **1** Selected **0** Not selected
		max		max="*nnn*" Integer Enter a number to be used as the maximum number of target database records to be selected per source database record for the listed logical and application relationships. If you do not specify, the relcrit max value is used if specified. If both are unspecified then there is no limit. It is recommended that a value of 2 or 3 is used for bidirectional logical relationships.
2	cset	subset	IMSCRIT	Criteria set subset number.
		sel	IMSCRIT	Boolean. Values: **1** Selected **0** Not selected
		desc	IMSCRIT	Optional description.
2	copybooks		BASE IMS	Group element for copybook definitions.
3	library			Data set name where a copybook can be found. This element can be repeated up to 12 times.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
3	syslib			Data set name where nested copybooks can be found. This element can be repeated up to ten times.
3	member	name		Copybook member name.
		lib		This specifies the library element number from which this copybook must be taken. If not specified the first occurrence in the library data sets specified.
		name01		01 name to be inserted by File Manager.
		segname	IMS only	Associated IMS segment name.
4	segdesc		IMS only	Segment description.
4	redefine	offset		Boolean. Set to "1" to tell File Manager to automatically adjust the offset value for each layout so that the starting location is the redefines, union or org field start location.
		chglvl		Boolean. Set to "1" if required. COBOL level change. This is a COBOL ONLY option which changes the way the level value (specified below) is processed. If you set this option then every occurrence of the specified level value is changed to 01 before the compile step, irrespective of whether the data item has a redefines clause. This option should be used with care as it may result in incorrect offsets to data items, or compile errors as the structure is changed from its intended programmable form.
		level		Source level value that is used to identify redefines or union level clauses that are to generate new record layouts. This value is used when you have a structure that has defined multiple layouts using COBOL redefines or PL/I unions. File Manager creates a separate record layout or each redefines clause at the given level. After the first matching redefines is found, File Manager creates new layouts for the next and subsequent redefines for the same level and start location. Header and trailing data items are included in each record layout. This field is ignored for Assembler copybooks.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		name		Field Name: This specifies the target field name of the redefines (COBOL), Union (PL/I) or ORG (Assembler) clause. File Manager uses this value to determine the redefines, union or org statements that are to be represented in a new layout. This is an alternative way of identifying redefines, union and org statements that require new layouts to be generated. You can specify both Level and Field name values and File Manager checks both when generating new layouts.
4	sourcerange	fromstr		From string. File Manager searches each source statement for the specified string. The first statement with a matching string starts the extract process. If the From statement is specified (not the default value), then it is used if the string is not found first.
		tostr		To string. File Manager searches each source statement for the specified string. The first statement with a matching string ends the extract process. If the To statement is specified, then the first statement that matches either the To statement value or contains the To string ends the extract.
		fromstmt		From Statement: Start source line number. This tells File Manager the starting statement for extract. If not specified the default start location is statement 1.
		tostmt		To Statement: End source line number. This tells File Manager the ending statement for extract. If not specified the default end location is the last source statement.
3	cobol	dbcs	BASE IMS	Boolean. Set `DBCS="1"` if the DBCS COBOL compiler option is required.
		maxrc		Maximum acceptable compiler RC, 0-20.
		mixedcase		Boolean. Set `mixedcase="1"` to retain the original case of the field name as coded in the COBOL copybooks. This feature is only available if you are running with the File Manager COBOL compiler or a minimum compiler level of Enterprise COBOL V4R1.
		dpc		Boolean. Set `dpc="1"` to use the Decimal-point is comma SPECIAL-NAMES paragraph when compiling COBOL copybooks.
		arith		Boolean. Set `arith="1"` to use the Arith(extend) COBOL compile option when compiling COBOL copybooks.
4	replace			Group element to provide COBOL replacing strings.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
5	repfrom			COBOL replace from string. There should be a matching repto element. Up to five replace from elements supported
5	repto			COBOL replace to string. Should be one for every repfrom string specified.
4	cbladdopp			Additional COBOL compiler options which is added via the CBL statement when a COBOL compile is run to build or update a template. These options are validated during the compile process. To avoid compile errors please ensure the syntax is correct and that any additional data sets required by these options are allocated prior to invoking File Manager. Do not exceed 50 bytes. `<cbladdop>compiler options</cbladdop>`
3	pli	graph	BASE IMS	Boolean. Set `graph="1"` to use the GRAPHIC PL/I compiler option.
		maxrc		Maximum acceptable compiler RC: 0-20.
		bin63		Boolean. Set `bin63="1"` to use the LIMITS(FIXEDBIN(63)) PL/I compiler option.
		fixdec		Boolean. Set `fixdec="1"` to use the LIMITS(FIXEDDEC(31)) PL/I compiler option.
		unalign		Boolean. Set `unalign="1"` to add the statement DEFAULT RANGE(*) UNALIGNED;. This changes the default for structure alignments.
4	pliaddopp			Additional PL/I compiler options which are added via the *PROCESS statement when a PL/I compile is run to build or update a template. These options are validated during the compile process. To avoid compile errors please ensure the syntax is correct and that any additional data sets required by these options are allocated prior to invoking File Manager. Do not exceed 50 bytes. `<pliaddop>compiler options</pliaddop>`
3	asm	dbcs	BASE IMS	Boolean. Set `dbcs="1"` to use the DBCS assembler compiler option.
		noalign		Boolean. Set `noalign="1"` to use the NOALIGN assembler compiler option.
4	asmaddopp			Additional HLASM compiler options which are added via the *PROCESS statement when a assembler is run to build or update a template. These options are validated during the assembly process. To avoid assembly errors please ensure the syntax is correct and that any additional data sets required by these options are allocated prior to invoking File Manager. Do not exceed 50 bytes. `<asmaddop>compiler options</asmaddop>`

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
2	layout	name	Not DB2	01 layout name - not applicable to DB2 templates
		copybook	BASE IMS	The name of the copybook where this 01 layout originated.
		sel	Not DB2	Boolean. The default value is “1”. Specify `sel="0"` to deselect the layout.
		segment	IMS IMSVIEW IMSCRIT	Segment name
		segsel	IMS IMSVIEW IMSCRIT	Boolean. The default value is “1”. This is segment selection - it only has value when used for Views and Criteria sets - Typically you can use this to turn off segment selection by setting `segsel="0"`.
		offset	not DB2	`offset="nnnn"` The default value is 0. The start location of all fields in the layout is adjusted by the value provided, which must be in the range -32760 to 32760.
3	criteria	type	Not DB2	Specify: **type="ID"** Identification criteria **type="SEL"** Selection criteria **type="RID"** Related identification criteria
		byfield		Boolean. The default value is “0”. Specify `byfield="1"` if expressions are specified using byline elements as opposed free format. <exp>expressions</exp> values are ignored if you specify `byfield="1"`.
		or	segmented	Boolean. The default value is “0” Specify `or="1"` to OR this criteria with related 01 criteria.
		rname	segmented	The name of related 01 layout. This is only valid if `type="RID"` is also specified.
		cset	IMSCRIT	`cset=nnn` The subset number this criteria belongs to. The subset number is defined on the cset element subset attribute described above.
4	exp		ALL	This is a free format expression or a where clause for DB2 templates. Note: As expressions generally contain special XML characters it is advisable to specify using CDATA as follows: `<exp><![CDATA[ my expression]]></exp>`

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
4	byline		ALL	Use the byline and child elements to describe your expression by field. The elements described here map to the by field representation entered through the ISPF interface to provide an expression by field. `byfield="1"` must be specified on the layout to specify these elements.
		seq		This is the sequence number starting from 1 and incrementing by 1 for each subsequent byline.
		lparen		Specify `lparen="("` to insert a left parenthesis before the expression defined by this element.
		rparen		Specify `rparen=")"` to insert a right parenthesis after the expression defined by this element
		conn		How this expression is to be connect to subsequent expressions. Specify `conn="AND"` or `conn="OR"`.
		name		The name of the field this expression is to use as the left hand side operand.
5	oper		not DB2	The arithmetic relational operation. See the operators described for the dynamic template panel in this manual for a full description. As operators generally contain special XML characters it is advisable to specify using CDATA as follows: `<oper><![CDATA[ < ]]></oper>`
	oper2		DB2	This is the DB2 operator as described in the *FM/DB2 Users Guide*, see "Row Selection Criteria" and the supported operators. As operators generally contain special XML characters it is advisable to specify using CDATA as follows: `<oper2><![CDATA[ <]]></oper2>`
	byval		ALL	This specified the right hand or second operand which must be a valid value for the specified operator and field or column referenced. It must not exceed 255 characters.
3	symbol		all	This element, with its attributes and child elements are used to specify the formatting, create, scrambling and for dynamic templates the field definition itself.
		name		This is the field or column name.
		sel		Boolean. The default value is "1". Specify `sel="0"` to deselect the column or field.
		seq	not DB2	This is a number representing the display order. Lowest number appears first. For example, `seq="1"`.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		hold		Boolean. The default value is "0". Specify `hold="1"` if you want this field or column to be held in a edit, view or browse session.
		lzero		Boolean. The default value is "0". Specify `lzero="1"` for numeric fields where the formatting is to include leading zeros
		width		Numeric. This specified an alternate display width for "TABL" formatting of the data. For example, `width="10"` sets a display width of 10.
		start	dynamic	Defines the start column for a field definition in a dynamic template.
		length	dynamic	Defines the length for a field definition in a dynamic template
		type	dynamic	Defines the type of field. See the types described for the dynamic template panel in this manual for a full description.
		keyseq		Integer. `keyseq='nnnn'` Specify this value to provide a key sequence number used to define the composite key field for data set comparison.
		db2ord	DB2	Integer. `db2ord='nnnn'` This is a sequence number for DB2 templates that indicates how to order the data based on the data values for the column, when the data is retrieved from DB2.
		db2AD	DB2	Integer. Ascending or Descending indicator. (`db2AD="A"`) This is only valid when db2ord is specified and indicates whether the data retrieved from DB2 is to be in A(ascending) or D(descending) order.
		ref		Integer. This is the field or column reference number and is used to identify the field. **Note:** A field can also be identified by the `name="field name"`. If this value is provided it takes precedence in identifying the field or column.
4	heading		ALL	Alternate heading `<heading>Myfield name</heading>`
4	lenfld		segmented	Tells us that the associated parent symbol is a length field for determining the length of the actual segment. This element should only be provided once for any given layout.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		excl		Boolean. Default value is “0”. Specify `excl="1"` to say the length of the segment is the length provided in this field plus the length of the field. If not specified or “0” then the length of the segment is the length value provided in this field.
4	createN	start	ALL	The default value is 0. `Start="nnnn"`, where nnn is numeric. Specifies the initial value you want a field to contain, before being adjusted by the increment you specify. The value must be a number that, when converted to the appropriate numeric data type, the field can hold, and, for a field with decimal places, must not specify more decimal integers than the number of decimal places in the field definition. If you specify a negative number, the sign is honored, even if the field is unsigned. If the field is a floating-point field, the start value can be specified as a floating-point number consisting of a mantissa and an exponent (such as -1.14579E01). The mantissa consists of an optional sign (+ or -) followed by 1 to 16 digits. The mantissa can also contain a decimal point. The exponent consists of the letter E, an optional sign (+ or -), and 1 or 2 digits.
		end		The default value is the largest positive or negative number the field can contain. `end="nnnn"` Specifies the maximum value (if the increment is a positive number) or minimum value (if the increment is a negative number) you want a field to contain. See rules for start attribute in specifying the number.
		inc		The default value is 0. `inc="nnnn"` Specifies a positive or negative number by which you want the value in the field adjusted for each record (or cycle of records). See rules for start attribute in specifying the number.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		cycle		The default value is 1. `cycle="nnnn"` Specifies the number of output records that are to be generated before the increment value is applied to the field value. For example, if you specify a field start value of 100, an increment value of 10, and a cycle value of 3, the field in the first three records contains 100, 110 in next three records, 120 in the next three records, and so on. If cycle is zero, the value in the field is always set to the start value.
4	createDT	value	DB2	`createdt="dd/mm/yy etc"` Specify a date value to be used when creating a new row. The creation attribute is only used for those functions where FM/DB2 is to create rows, for example the create utility. You can specify a date value in the same format shown in the Format field. You can specify an = to indicate that the current date is to be used. When you specify this the current value is retrieved and replaces the = in the field. The value retrieved is a constant, so that if a create utility is run later, the same value is always used to populate the column. You can specify * to indicate that the current date when the row is created should be used to populate the column. Specifying * indicates a variable value for the column when a create utility is run, since the value is the date on which the create utility is executed.
		inc		`inc="nnnn"` Specify an increment value to be used when creating multiple rows. Valid values are integers. A value of 1 indicates that each subsequent row should have a date value one day later than the previous date value. A value of -1 indicates that each subsequent row should have a date value one day earlier than the previous date value.
4	createC		ALL	Use this to populate field with various patterns.
		act		Action. Specifies how the contents of the field are to be varied for each record. For more information. For example, `act="FX"`. For details of each action, see “Options for "act"” on page 1211.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		repeat		The default is NO. `repeat="YES"` Repeat user pattern. Specify YES if you have provided a user-supplied pattern that is shorter than this field and you want the pattern to be repeated as many times as necessary to fill the field. By default, File Manager uses the fill character to pad the receiving field when the user-supplied pattern is shorter than the field. This option has no effect when the RA action is specified. IBM-supplied patterns are always repeated in a field, as necessary.
5	filler		ALL	`<filler>cccc</filler>` Specifies a value that is to be placed in each byte of the field before any other operation in the construction of the field. Can be one of: **char** A character such as 0 to be written in each byte. **X'cc'** A hexadecimal value, such as X'FF' to be written in each byte.
5	start			`<start>cccc<start>` Start character Sets the starting character to be used when you specify an IBM-supplied pattern (AL, AN, or CO) or a user-supplied pattern (with the exception of RO, WV, and FX). The specified character must be one of the characters in the IBM-supplied pattern or user-supplied pattern. Default: First character in IBM-supplied pattern or user-supplied pattern.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
5	pattern			`<pattern>cccc<pattern>` Specifies the pattern to be used when generating data for this field. You can specify either an IBM-supplied pattern indicator or a user-supplied pattern. The user-supplied pattern can be either a character string or a hexadecimal string. A character string must be enclosed in quotes, while a hexadecimal string must be enclosed in quotes and be preceded by a X. The hexadecimal string must contain an even number of valid hexadecimal characters. The IBM-supplied patterns you can specify are: **AL** Alphabetic - characters A-Z, inclusive. **AN** Alphanumeric - characters A-Z, 0-9 inclusive. **AX** Alphanumeric displayed in long hexadecimal. **CO** Collating sequence - all printable characters in the range X'40' - X'F9'. If you provide a user-supplied pattern that is longer than the field, for actions other than RA and RP the pattern is first truncated on the right to fit the receiving field before the specified action is performed. For the RA action, characters are randomly selected from the entire user-supplied pattern. For the RP action, the entire pattern is rippled for each record before it is truncated to fit the receiving field. If you provide a user-supplied pattern that is shorter than the field, you can specify that you want the pattern to be repeated to fill the field. Default: None
4	scramble		ALL	Use the scramble element to specify all the scrambling type options for the associated parent symbol.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		type		For example, `type="1"` Scramble Type. Specify one of these values: **Blank** No scrambling is performed. Value or range specifications are saved but ignored for the associated function. **1 (Random)** Performs random scrambling. The same input value produces different output values on subsequent invocations. **2 (Repeatable)** Performs repeatable scrambling. The same input value produces the same output value on subsequent invocations. **3 (Translate)** Performs translation. The value data set is searched to find a matching input value. If a match is found, then the output value is taken from the output column of the matching record. **4 (Exit)** Invokes a scrambling user exit. File Manager displays a panel where you can specify the user exit name and user exit parameters and options.
5	translate	incol		incol="nnn" Defines the start location of the input field value on the value data set, and is used when the translate process is run during a copy operation to match the input field with a value on the value data set. The length of the field is set to the length of the input field that is mapped to this field during the copy process.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		outcol		outcol="nnn" Defines the start location of the output field value on the value data set, and is used as follows during a copy operation: **For scramble type Translate** If an input field value is matched on the value data set, then the corresponding output value is used. **For scramble types Random or Repeatable** The input value is used to randomly or repeatably select an output value from the value data set. The length of the field is the current field length as displayed on this panel. This value is required when you select the scramble type Translate. If you select Random or Repeatable, and also select Value, then the start location defaults to 1 if a value data set name (Dsn) has been provided.
		dsn		dsn="data set name" Defines the value data set. It can be any catalogued sequential, partitioned or VSAM data set containing data that is used to determine the output field value during a copy process.
5	range			This restricts scrambling values to a range of numbers between the min and max values provided on the attributes. You must provide a min and max value
		min		Minimum value. Must be less than the maximum value and greater than or equal to -2GB.
		max		Maximum value. Must be greater than the minimum value and less than or equal to 2GB
5	value	outcol		`outcol="nnn"` Defines the start location of the output field value on the value data set, and is used as follows during a copy operation: **For scramble type Translate** If an input field value is matched on the value data set, then the corresponding output value is used. **For scramble types Random or Repeatable** The input value is used to randomly or repeatably select an output value from the value data set.

Table 21. XML definitions for a template (continued)

Lvl	Element	Attribute	Type	Description
		dsn		`dsn="data set name"` Defines the value data set. It can be any cataloged sequential, partitioned or VSAM data set containing data that is used to determine the output field value during a copy process.
6	sval			`<sval>field value</sval>` The values you provide are randomly or repeatably selected depending on your scrambling option to populate the output field during the scrambling operation. The field value must be valid data for the associated parent symbol.
5	exitprog	name	all	Enter a 1 to 8 character valid load module member name to identify the scramble exit that is invoked during a copy operation for this output field. Any exit you provide must be in the form of a load module, in any load library available to File Manager at the time of the copy process, either by a STEPLIB DD statement, or in LINKLIST, or LPALIST.
		parm		You can specify a constant value that is passed to the exit for each call type by entering a non blank value in this field.
		format		Boolean. Default value is "0". Specify `format="1"` to format the field value passed to the exit as display numeric with leading zeros suppressed.
		lzero		Boolean. Default value is "0". Specify `lzero="1"` to format the field value passed to the exit as display numeric with leading zeros.

Options for "act"

Here are the possible values for "act" (which specifies how the contents of the field are to be varied for each record):

FX The contents of the field are to remain fixed.

RA The contents of the field are to be generated from characters selected at random from the specified pattern. A different set of characters is selected for each record.

RO The contents of the field are to be rolled. The pattern you specify is shifted left one byte for each record until the first non-blank character of the pattern is in the first byte of the field. Then, the pattern is shifted right one byte for each output record until it returns to its original position in the field. RO can only be used with a user-supplied pattern, not with an IBM-supplied pattern.

The pattern must start with a blank, or the result is the same as FX. The roll only occurs within the length of the pattern.

RP The contents of the field are to be rippled. The pattern you specify is shifted left one byte for each record and the truncated character is appended to the right hand end of the pattern.

SL The contents of the field are to be shifted left for each record. The pattern you specify is shifted left one character and filled to the right with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

SR The contents of the field are to be shifted right for each record. The pattern you specify is shifted right one character and filled to the left with a space. When all characters in the pattern have been shifted out, the original pattern is restored and the process is repeated.

TL The contents of the field are to be truncated on the left for each record. The pattern you specify is left truncated (the leftmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then, the original pattern is restored and the process is repeated.

TR The contents of the field are to be truncated on the right for each record. The pattern you specify is right truncated (the rightmost character replaced with a space) one character at a time until all characters in the pattern have been truncated. Then, the original pattern is restored and the process is repeated.

WV The contents of the field are to be waved. The pattern you specify is shifted left one byte for each record until the first non-blank character of the pattern is in the first byte of the field. Then, the original pattern is restored and the process repeated. WV can only be used with a user-supplied pattern, not with an IBM-supplied pattern.

The pattern must start with a blank, or the result is the same as FX. The roll only occurs within the length of the pattern.

Support resources and problem solving information

This section shows you how to quickly locate information to help answer your questions and solve your problems. If you have to call IBM support, this section provides information that you need to provide to the IBM service representative to help diagnose and resolve the problem.

For a comprehensive multimedia overview of IBM software support resources, see the IBM Education Assistant presentation "IBM Software Support Resources for System z Enterprise Development Tools and Compilers products" at http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp?topic=/com.ibm.iea.debugt/debugt/6.1z/TrainingEducation/SupportInfoADTools/player.html.

Searching IBM support Web sites for a solution

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

You can find this publication and documentation for many other products in the IBM System z Enterprise Development Tools & Compilers information center at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp. Using the information center, you can search product documentation in a variety of ways. You can search across the documentation for multiple products, search across a subset of the product documentation that you specify, or search a specific set of topics that you specify within a document. Search terms can include exact words or phrases, wild cards, and Boolean operators.

To learn more about how to use the search facility provided in the IBM System z Enterprise Development Tools & Compilers information center, you can view the multimedia presentation at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp?topic=/com.ibm.help.doc/InfoCenterTour800600.htm.

Searching product support documents

Use the System z Enterprise Development Tools & Compilers information center or the IBM support site at www.ibm.com/software/support to search for the latest, most complete information that might help you resolve your problem.

When you access the IBM support site, you can specify any of the following products for which you want information to be displayed:

- Application Performance Analyzer for z/OS
- Debug Tool for z/OS
- Enterprise COBOL for z/OS
- Enterprise PL/I for z/OS
- Fault Analyzer for z/OS
- File Manager for z/OS
- Optim Move for DB2
- WebSphere Developer Debugger for System z
- Workload Simulator for z/OS and OS/390 Support

When you access the IBM support site, you can also use the IBM Support Portal to customize the support information to be displayed and save product names that you specify. There is also a search facility provided with the IBM Support Portal that allows you to narrow the search scope and search only product support documents for the products that you specify. The IBM Support Portal can be accessed through the IBM support site at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal. For information about customizing your IBM support site experience using the IBM Support Portal, refer to https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos?lang=en_us.

IBM Support Assistant

The IBM Support Assistant (also referred to as ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. It provides quick access to support-related information. You can use the IBM Support Assistant to help you in the following ways:

- Search through IBM and non-IBM knowledge and information sources across multiple IBM products to answer a question or solve a problem.
- Find additional information through product and support pages, customer news groups and forums, skills and training resources and information about troubleshooting and commonly asked questions.

In addition, you can use the built in Updater facility in IBM Support Assistant to obtain IBM Support Assistant upgrades and new features to add support for additional software products and capabilities as they become available.

For more information, and to download and start using the IBM Support Assistant for IBM System z Enterprise Development Tools & Compilers products, please visit http://www.ibm.com/support/docview.wss?rs=2300&context=SSFMHB&dc=D600&uid=swg21242707&loc=en_US&cs=UTF-8&lang=en.

General information about the IBM Support Assistant can be found on the IBM Support Assistant home page at http://www.ibm.com/software/support/isa.

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes and other updates are available, the following information is available from the IBM support site. You can also view the following information from the IBM Support Portal when you specify the applicable products.

- Latest PTFs for Application Performance Analyzer for z/OS
- Latest PTFs for Debug Tool for z/OS

- Latest PTFs for Fault Analyzer for z/OS
- Latest PTFs for File Export for z/OS
- Latest PTFs for File Manager for z/OS
- Latest fixes for Optim Move for DB2
- Latest PTFs for WebSphere Studio Asset Analyzer for Multiplatforms
- Latest PTFs for Workload Simulator for z/OS and OS/390

When you find a fix that you are interested in, click the name of the fix to read its description and to optionally download the fix.

The IBM Support Portal is a way for you to specify specific products for which you want to display support information. The Support Portal can be accessed through the IBM support site at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal. For information about how to customize your IBM support site experience using the IBM Support Portal, refer to https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos?lang=en_us.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/handbook.html.

Receiving support updates through e-mail notification

To receive e-mail notifications about fixes and other software support news, follow the steps below. Additional information is provided at http://www.ibm.com/support/docview.wss?rs=615&uid;=swg21172598.

1. Go to the IBM software support site at http://www.ibm.com/software/support.
2. Click **Request e-mail updates** in the **Additional support links** section of the page.
3. Click any **My Notifications** link on the page that is displayed.
4. If you have already registered for **My notifications**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
5. In the **My notifications** tool, click the **Subscribe** tab to specify products for which you want to receive e-mail updates.
6. To specify Problem Determination Tools products, click **Other software** and then select the products for which you want to receive e-mail updates, for example, **Debug Tool for z/OS** and **File Manager for z/OS**.
7. To specify a COBOL or PL/I compiler, click **Rational®** and then select the products for which you want to receive e-mail updates, for example, **Enterprise COBOL for z/OS** .
8. After selecting all products that are of interest to you, scroll to the bottom of the list and click **Continue**.
9. Determine how you want to save your subscription. You can use the default subscription name or create your own by entering a new name in the **Name** field. It is recommended that you create your own unique subscription name using a something easily recognized by you. You can create a new folder by entering a folder name in the **New** field or select an existing folder from the pulldown list. A folder is a container for multiple subscriptions.
10. Specify the types of documents you want and the e-mail notification frequency.

11. Scroll to the bottom of the page and click **Submit**.

To view your current subscriptions and subscription folders, click **My subscriptions**.

If you experience problems with the **My notifications** feature, click the **Feedback** link in the left navigation panel and follow the instructions provided.

Receiving support updates through RSS feeds

To receive RSS feeds about fixes and other software support news, go to the following web site and select the products in which you are interested:

- http://www.ibm.com/software/support/rss/other/index.html.

If you need to contact IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/ WebDocs/Passport_Advantage_Home and click **How to Enroll**.

  By phone

  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, S/390®, iSeries®, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determining the business impact"
2. "Describing problems and gathering information"
3. "Submitting problems" on page 1218

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
: The problem has a **critical** business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
: The problem has a **significant** business impact. The program is usable, but it is severely limited.

Severity 3
: The problem has **some** business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
: The problem has **minimal** business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently.

To save time, if there is a Mustgather document available for the product, refer to the Mustgather document and gather the information specified. Mustgather documents contain specific instructions for submitting your problem to IBM and gathering information needed by the IBM support team to resolve your problem. To determine if there is a Mustgather document for this product, go to the product support page and search on the term Mustgather. At the time of this publication, the following Mustgather documents are available:

- Mustgather: Read first for problems encountered with Application Performance Analyzer for z/OS: http://www.ibm.com/support/docview.wss?rs=2300 &context=SSFMHB&q1=mustgather&uid=swg21265542&loc=en_US&cs=utf-8 ⟨=en
- Mustgather: Read first for problems encountered with Debug Tool for z/OS: http://www.ibm.com/support/docview.wss?rs=615&context=SSGTSD &q1=mustgather&uid=swg21254711&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Fault Analyzer for z/OS:http://www.ibm.com/support/docview.wss?rs=273&context=SSXJAJ &q1=mustgather&uid=swg21255056&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with File Manager for z/OS: http://www.ibm.com/support/docview.wss?rs=274&context=SSXJAV &q1=mustgather&uid=swg21255514&loc=en_US&cs=utf-8&lang=en

- Mustgather: Read first for problems encountered with Enterprise COBOL for z/OS: http://www.ibm.com/support/docview.wss?rs=2231&context=SS6SG3&q1=mustgather&uid=swg21249990&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Enterprise PL/I for z/OS: http://www.ibm.com/support/docview.wss?rs=619&context=SSY2V3&q1=mustgather&uid=swg21260496&loc=en_US&cs=utf-8&lang=en

If the product does not have a Mustgather document, please provide answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Open service request** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. In the Other support tools section, select IBMLink to open an Electronic Technical Response (ETR). Enter your information into the appropriate problem submission form.

By phone

Call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

After a Problem Management Record (PMR) is open, you can submit diagnostic MustGather data to IBM using one of the following methods:

- FTP diagnostic data to IBM
- If FTP is not possible, email diagnostic data to techsupport@mainz.ibm.com. You must add PMR xxxxx bbb ccc in the subject line of your email. xxxxx is your PMR number, bbb is your branch office, and ccc is your IBM country code. Click here http://itcenter.mainz.de.ibm.com/ecurep/mail/subject.html for more details.

Always update your PMR to indicate that data has been sent. You can update your PMR online or by phone as described above.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA
95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: ©(your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. ©Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

The User's Guide and Reference documents intended Programming Interfaces that allow the customer to write programs to obtain the services of File Manager.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information”, http://www.ibm.com/legal/copytrade.shtml.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Bibliography

File Manager publications

File Manager Customization Guide, SC19-4118
File Manager User's Guide and Reference, SC19-4119
File Manager User's Guide and Reference for DB2 Data, SC19-4120
File Manager User's Guide and Reference for IMS Data, SC19-4121
File Manager User's Guide and Reference for CICS, SC19-4122
File Manager Fact Sheet, G325-2429
File Manager License Information, GC19-4117
File Manager Program Directory, GI10-8968

Related publications for Problem Determination Tools

IBM Problem Determination Tools for z/OS Common Component Customization Guide and User Guide, SC19-4159

Related publications for COBOL

IBM COBOL Language Reference, SC26-9046
IBM COBOL Programming Guide for OS/390 & VM, SC26-9049

Related publications for PL/I

IBM VisualAge PL/I Language Reference, SC26-9476
IBM VisualAge PL/I for OS/390 Programming Guide, SC26-9473

Related publications for z/OS

z/OS DFSMS Access Method Services for Catalogs, SC26-7394
z/OS DFSMS Object Access Method Application Programmer's Reference, SC35-0425
z/OS DFSMS: Using Data Sets, SC26-7410
z/OS DFSMS: Using Magnetic Tapes, SC26-7412
z/OS ISPF User's Guide Vol 1, SC34-4822
z/OS ISPF User's Guide Vol II, SC34-4823
z/OS MVS JCL Reference, SA22-7597
z/OS MVS System Messages, Vol 5, SA22-7635
z/OS Support for Unicode Using Conversion Services, SA22-7649
z/OS TSO/E Command Reference, SA22-7782
z/OS TSO/E Programming Services, SA22-7789
z/OS TSO/E REXX Reference, SA22-7790
z/OS TSO/E REXX User's Guide, SA22-7791

Glossary

This publication includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing*, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

The following terms are defined as they are used in this manual. If you do not find the term you are looking for, refer to the Index or to the *IBM Dictionary of Computing*.

ANSI definitions are preceded by an asterisk.

A

across format
A File Manager dump format that has the same appearance as a system dump. To get this format for your listings, use the SET DUMP=ACROSS processing option.

action bar
The area at the top of a window that contains choices that give a user access to choices available in that window.

ASCII * American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

B

block * A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps and each block may contain one or more logical records.

A block is a *physical record* but it may be composed of several contiguous *logical records* or parts of *logical records*.

blocking
The process of combining two or more *logical records* in one block.

BLP Bypass label processing.

bounds
In editor functions, the default *column range* within which commands operate. In editor functions, the bounds can be changed using the BOUNDS command.

buffer A portion of storage used to hold input or output data temporarily.

C

cards Lines in a batch job.

cartridge
A storage device that consists of magnetic tape, on supply and takeup reels, in a protective housing.

catalog
* A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, and blocking factors. * To enter information about a file or library into a catalog.

character fill
* To insert as often as necessary into a storage medium the representation of a specified character that does not itself convey data but may delete unwanted data.

CKD Count key data, a disk architecture.

collection
See *object collection*.

column range
In editor functions, the left and right columns that control which data is processed by edit commands. Some editor commands allow you to specify a column range to override the *bounds*.

console
See *operator console*.

control interval
A fixed-length area of direct access storage in which VSAM stores records

and creates distributed free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always comprises an integral number of *physical records*.

copybook
In general, a copybook is a file containing a sequence of code that is included in a source program at compile time. In File Manager, the term "copybook" refers specifically to a file containing descriptions of record structures.

cylinder
The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

D

data definition (DD) statement
A job control statement describing a data set associated with a specific job step.

data definition name (ddname)
The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

data-in-virtual data set
A data set that is considered a continuous stream of uninterrupted data.

DBCS Double-byte character set.

deblocking
The process of making each *logical record* of a block available for processing. Contrast with *blocking*.

E

EBCDIC
* Extended Binary-Coded Decimal Interchange Code. A coded character set consisting of 8-bit coded characters.

end of file (EOF)
An internal label or special record immediately following the last record of a file, signaling the end of that file. It may include control totals for comparison with counts accumulated during processing.

end of volume (EOV)
An internal label, or 2 consecutive tape marks, that follows and indicates the end of a set of data contained in a volume.

excluded record
In edit functions, a record that has been excluded from display using the EXCLUDE *primary command* or the X *prefix command*.

extended help
Help text that applies to an entire function or panel. Contrast with *field help*.

extent Continuous space on a disk that is occupied by or reserved for a particular data set, data space, or file.

F

field help
Help text that applies to a specific entry field, parameter, or term. Contrast with *extended help*.

field mapping
The relationship between fields in the "from" data set and fields in the "to" data set. Field mapping defines which fields in the "from" data set are copied to which fields in the "to" data set. The field mapping is defined using the Template Workbench.

field list
When data is displayed in TABL or SNGL display format, specifies which fields an editor command is to operate on.

file, tape
See *tape file*.

fill See *character fill*.

full-screen mode
A type of File Manager session, in which you can use full-screen panels to select functions, specify parameters, view data, and update data.

H

header label
An internal label, immediately preceding the first record of a file, that identifies the file and contains data used in file control.

I

interactive (processing)
Pertaining to an application in which each entry calls for a response from a system or program, as in an inquiry system or an airline reservation system. An interactive

system may also be conversational, implying a continuous dialog between the user and the system.

K

KSDS Key-sequenced data set.

L

label A record that identifies a volume on tape or disk or that identifies a file on the volume.

label, header
See *header label*.

label, record
See *record label*.

line command
In full-screen mode, a command that you can enter on any *list panel* by typing the command over the beginning of a list entry.

linear data set
A named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

list panel
In full-screen mode, a panel that displays a list of entries such as data sets, members, or objects, and that enables you to perform functions on one or more of the entries.

logical record
A record from the standpoint of its content, function, and use rather than its physical attributes, that is, a record defined in terms of the information it contains. In VSAM, a unit of information normally pertaining to a single subject; a logical record is the user record requested of or given to the data management function.

M

multiple record types
Records in a single file that are of different lengths or contain different types of data.

N

not-selected record
A record that has not been selected for processing by the function because it does not meet the record identification criteria or the record selection criteria.

object In OAM, data consisting of a contiguous stream of bytes of arbitrary length, up to a maximum of 15 megabytes. The contents and internal structure of an object are defined by the application using it. Objects are increasingly used in image processing and other advanced application areas.

object collection
A collection of user-defined objects.

operator console
A functional unit containing devices that are used for communications between a computer operator and a computer.

O

OAM Object Access Method.

P

parameter string
Two or more parameters separated by commas or blanks.

PDS Partitioned data set.

physical record
A record whose characteristics depend on the manner or form in which it is stored, retrieved, or moved. A physical record may contain all or part of one or more *logical records*.

prefix commands
In full-screen mode, commands you enter in the prefix area of the panel. Prefix commands are used by File Manager edit functions.

primary command
In full-screen mode, commands you enter on the Command line. File Manager provides common primary commands that are available across a range of functions, and function-specific primary commands, such as editor commands.

Q

QSAM
Queued Sequential Access Method, an access method for sequential data sets.

R

record identification criteria
Criteria that are used to identify different record types in a file. The two types of record identification criteria are implicit and explicit. The default (implicit) record

identification criteria are the minimum and maximum lengths of records of a given record type. Explicit record identification criteria are criteria you specify using the Template Workbench.

record label
In edit functions, an alphabetic character string used to name a particular record for easy reference.

record label range
In edit functions, a pair of *record labels* specified in an edit command to define the records on which the command is to operate.

record selection criteria
Criteria that are used for selecting records for processing by a function, and specified using the Template Workbench.

record size
The number of characters or bytes in a record.

record type
The set of records in a single file that are the same length or contain the same type of data.

RRDS Relative record data set.

S

SAF System Authorization Facility, a system facility for security.

shadow line
In editor functions, a line that indicates the position of records that have been excluded from display. A shadow line can represent *excluded records*, *not-selected records*, or *suppressed records*.

shift-out/shift-in
Characters that mark the beginning and end of double-byte character set data. The shift-out character is X'0E', and the shift-in character is X'0F'.

SMS Storage Management Subsystem.

standard label
A fixed-format record that identifies a tape that is part of a multivolume tape file.

suppressed record
In editor functions, a record that has been selected for processing by the function, but is of a different record type to the currently displayed record type.

T

tape file
One or more data blocks between tape marks.

tape mark
An indicator record that marks the beginning or the end of a set of data such as a file.

tape unit
A device containing a magnetic tape drive, magnetic heads, and associated controls.

tape volume
A reel of magnetic tape.

template
A set of information, created using the *Template Workbench*, that describes the format of records in a file, how different record types are identified, what records and fields are to be selected for processing, how fields are to be presented when displayed or printed, how records are to be mapped when copied from one file to another, and how fields are to be initialized.

Template Workbench
A series of File Manager panels. With the Template Workbench you can specify the name of a copybook that describes your data, and you can create and edit a template that specifies which records and fields within your data set you want to process.

U

unit, tape
See *tape unit*.

updown format
A File Manager dump format with hexadecimal representation of data in 3 lines: character, zone, and number. To get this format for your listings, use the SET DUMP=UPDOWN processing option.

utility program
A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program.

V

volume
Tape volume: A reel of magnetic tape. Disk volume: A disk pack or part of a disk storage module.

volume serial number (volser)
A number in a volume label assigned when a volume is prepared for use in the system.

volume table of contents (VTOC)
A table on a direct access volume that describes each file on the volume.

VSAM
Virtual Storage Access Method.

Index

Special characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Printed in USA

SC19-4119-00