Interactive System Productivity Facility (ISPF)

User's Guide Volume I

z/OS Version 1 Release 2.0

SC34-4822-01

Interactive System Productivity Facility (ISPF)

User's Guide Volume I

z/OS Version 1 Release 2.0

SC34-4822-01

Note
Before using this document, read the general information under "Notices" on page 223.

Second Edition (October 2001)

This edition applies to ISPF for Version 1 Release 2 of the licensed program z/OS (program number 5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 284-4721.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for comments appears at the back of this publication. If the form has been removed, and you have ISPF-specific comments, address your comments to:

International Business Machines Corporation
Software Reengineering
Department G7IA / Building 503
Research Triangle Park, NC 27709-9990

FAX (United States & Canada): 1+800+227-5088
IBMLink (United States customers only): CIBMORCF@RALVM17
IBM Mail Exchange: USIB2HPD@VNET.IBM.COM
Internet: USIB2HPD@VNET.IBM.COM

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

Title and order number of this book
Page number or topic related to your comment

The ISPF development team maintains a site on the World-Wide Web. The URL for the site is:

http://www.software.ibm.com/ad/ispf

Contents

Figures

Tables

Preface

This book provides reference and usage information for programmers who develop applications with ISPF. It also provides conceptual and functional descriptions of ISPF.

About This Book

This book contains two parts. The first part provides introductory information about using ISPF:

- How to exploit the ISPF user interface
- An overview of ISPF
- How to use referral lists
- A description of ISPF commands
- How to use libraries and data sets
- How to prepare to run on MVS.

The second part contains the appendixes and provides:

- Information you need before you install the ISPF Client/Server component workstation interface
- APL and TEXT character conversion information
- Lists of abbreviations for commands, field values, keywords/operands, scroll amounts, and programming languages
- Descriptions of allocation data sets
- Descriptions and examples of the output listing formats.

Information about using ISPF Options is contained in the *ISPF User's Guide Volume II*.

Who Should Use This Book

This book is for application programmers using ISPF. Users should be familiar with coding CLISTs, REXX EXECs, or programs in the MVS environment.

What Is in This Book

Chapter 1. The ISPF User Interface, provides an overview of the ISPF user interface. This chapter also describes the Graphical User Interface (GUI); that is, how ISPF runs on a workstation. Information is provided on:

- Starting a GUI session
- What GUI mode looks like
- Some GUI restrictions
- Details for the Dialog Developers.

Chapter 2. Overview of ISPF, describes ISPF uses and the functions provided by the ISPF licensed program. In previous releases of ISPF, the information in this chapter was contained in the *ISPF/PDF Guide and Reference*.

Chapter 3. Using Personal Data Set Lists and Library Lists, describes the four types of referral lists, which are used to retrieve a stored data set or library to the currently displayed panel:

- Reference data set list
- Reference library list
- Personal data set list
- Personal library list.

Chapter 4. Using Commands, Function Keys, and Light Pen or Cursor Selection, describes ISPF system commands, the function keys and their default assignments, and the light pen and cursor select facilities. In previous releases of ISPF, the information in this chapter was contained in the *ISPF Dialog Management Guide and Reference*.

Chapter 5. Libraries and Data Sets, describes how to allocate, create, and use libraries and data sets. In previous releases of ISPF, the information in this chapter was contained in the *ISPF/PDF Guide and Reference*.

Chapter 6. Getting Ready to Run on MVS, helps you prepare to use ISPF data-element libraries. In previous releases of ISPF, the information in this chapter was contained in the *ISPF/PDF Guide and Reference*.

Appendix A. Installation Considerations for the Client/Server, provides information that you need to know before you begin to install the ISPF Client/Server component workstation software.

Appendix B. Configuring Communications for the ISPF Client/Server provides information that you need for configuring your communications protocols so you can use the ISPF Client/Server.

Appendix C. Listing Formats, describes and displays the kinds of listings you can produce using ISPF. The sample listings shown are for illustration purposes only. They are not intended to be exact replicas because printouts of ISPF listings vary according to the kind of printer you are using. In previous releases of ISPF, the information in this appendix was contained in the *ISPF Dialog Management Guide and Reference*.

Appendix D. APL and TEXT Character Conversion, describes how APL and TEXT characters are converted by ISPF for internal storage. In previous releases of ISPF, the information in this appendix was contained in the *ISPF/PDF Guide and Reference*.

Appendix E. Abbreviations for Commands and Other Values, lists commands, field values, keywords/operands, and scroll amounts that can be abbreviated. In previous releases of ISPF, the information in this appendix was contained in the *ISPF/PDF Guide and Reference*.

Appendix F. Allocation Data Sets, provides information on ALLOC commands that ISPF issues based on ISPF libraries, data set names, list IDs, options, and additional input libraries. In previous releases of ISPF, the information in this appendix was contained in the *ISPF/PDF Guide and Reference*.

Appendix G. ISRDDN Diagnostic Utility, provides information about ISRDDN, a program shipped with ISPF as a service aid.

Summary of Changes

z/OS V1R2.0 ISPF contains the following changes and enhancements:
- ISPF Product and Library Changes
- ISPF Dialog Manager Component Changes (including DTL changes)
- ISPF PDF Component Changes
- ISPF SCLM Component Changes
- ISPF Client/Server Component Changes

ISPF Product Changes

Changes to the ZENVIR variable. Characters 1 through 8 contain the product name and sequence number in the format *ISPF x.y*, where x.y indicates:
- <= 4.2 means the version.release of ISPF
- = 4.3 means ISPF for OS/390 release 2
- = 4.4 means ISPF 4.2.1 and ISPF for OS/390 release 3
- = 4.5 means ISPF for OS/390 Version 2 Release 5.0
- = 4.8 means ISPF for OS/390 Version 2 Release 8.0
- = 5.0 means ISPF for OS/390 Version 2 Release 10.0
- OR
- = 5.0 means ISPF for z/OS Version 1 Release 1.0
- = 5.2 means ISPF for z/OS Version 1 Release 2.0

The ZENVIR variable is used by IBM personnel for internal purposes. The x.y numbers DO NOT directly correlate to an ISPF release number in all cases. For example, as shown above, a ZENVIR value of 4.3 DOES NOT mean ISPF Version 4 Release 3. NO stand-alone version of ISPF exists above ISPF Version 4 Release 2 Modification 1.

The ZOS390RL variable contains the ISPF release on your system.

The ZISPFOS system variable contains the level of ISPF code that is running as part of the operating system release on your system. This might or might not match ZOS390RL. For this release, the variable contains **ISPF for z/OS 01.02.00**.

New system variables:

ZDAYOFWK
The day of the week.

The ISRDDN utility is now documented in the ISPF User's Guide.

ISPF DM Component Changes

The DM component of ISPF includes the following new functions and enhancements:
- Add support for "VER(&variable,IPADDR4)".
- Add the NOSETMSG parameter to the CONTROL Service.
- Add the LFORMAT parameter to the VDEFINE Service to allow defining like format variables in a list.
- Change tutorial processing to eliminate the "End of data" message on scrollable area panels that display the entire scrollable area on the screen (no More: + - is displayed). This change eliminates the extra enter the user had to execute before continuing to the next panel.

- Issue a TSO line message when a help panel is not found and continue the dialog. Previously ISPF issued a severe error message when a help panel could not be found.
- Display a message indicating a message is not found when running in Dialog Test and allow the dialog to continue.
- Add support for extended SBCS and DBCS CCSIDs:
  - 1159 Traditional Chinese
  - 1364 Korean
  - 1371 Traditional Chinese
  - 1388 Simplified Chinese
  - 1390 Japanese
  - 1399 Japanese
- Add new Z variables to support 5 character code pages and character sets, ZTERMCP5 and ZTERMCS5 respectively.
- Add new variable ZDAYOFWK to show the day of the week.
- Enhance the Reflist function of TEST option 7.6 to allow better list management.
- Enhance Locate and Find for Dialog Test Variables (option 7.3).
- A new exec called ISPCMDTB to convert ISPF command tables to DTL.
- A new Configuration Table variable to allow SCROLL defaults.
- A new Configuration Table variable to allow STATUS AREA defaults.

ISPDTLC enhancements:

ISPDTLC changes include new invocation options, new tags, and new tag. attributes as ISPF extensions to the Dialog Tag Language

General improvements:

- New invocation options:
  - no new invocation options in this release
- New tags:
  - DLDIV, DTDIV, DTHDIV for dividers within the DL tag
  - PLDIV, PTDIV for dividers within the PARML tag
- Replication added to predefined entities. For example, &GTSYM(5); will create the string '>>>>>' in the substituted text.
- National language text strings are now accessible as entities. For example, &command; will create the string 'Command' or its translated equivalent in the substituted text.
- New ENTITY keywords COPIES, X2C and ATTR.
- New macro tag default initialization processing syntax.

  ```
  <?dummy ?var=value>
  ```
- New Predefined ENTITY keywords cmdpmt (&cmdpmt;) and rptr (&rptr;).

New or changed tag attributes:

Tag name	Attribute update
CHECKI	Add support for "VER(&variable, IPADDR4)"
COMPOPT	Add ADD.

Tag name	Attribute update
DL	Add FORMAT. Support multiple DT tags for each DD tag. Change TSIZE to support multiple values. Each TSIZE value implies a DT tag.
DT	Add FORMAT, NOSKIP.
DTAFLD	Add AUTOTYPE, AUTOVOL, AUTODMEM.
HELP	Add ZUP, ZCONT.
Hn	Add COMPACT.
HP	Add INTENSE.
NOTE	Add NOSKIP.
NT	Add NOSKIP.
PANEL	Add ZUP, ZCONT, AUTONRET, AUTOTCMD.
PARML	Add FORMAT. Support multiple PT tags for each PD tag. Change TSIZE to support multiple values. Each TSIZE value implies a PT tag.
PT	Add FORMAT, SKIP.
SELFLD	Add SELCHECK.
	Support INIT=init-value for single-choice selection fields.

ISPF PDF Component Changes

The ISPF PDF component contains the following new functions and enhancements:

- A MEMBER command has been added to data set list (option 3.4) to allow the partitioned data sets in the list to be searched for a specific member.
- When the EDIT service is specified with an initial macro, parameters can now be specified for the initial macro.
- A FIND command has been added to member list to allow a string to be searched for in any of the displayed statistics.
- A SRCHFOR command has been added to data set list to allow SuperC to be invoked to search the listed data sets for strings.
- Move/Copy will now dynamically calculate the sized for the IEBCOPY SYSUT3 and SYSUT4 data sets.
- A QUERYENQ service has been added to retrieve ENQ information about a data set in use.
- LMF has been removed from the ISPF product.
- A new SuperC option FINDALL has been added to specify that all strings must be found to issue a ″strings found″ return code.
- LMPRINT will now allow the INDEX parameter to be specified for a record format U data set.
- Foreground and Batch now support the z/OS C/C++ compiler.

- A new AUTOTYPE command can be set to a PFKEY to retrieve a data set name or pattern entered on a panel based on data sets that start with that partial name.
- Data sets with an LRECL less than 10 bytes can be edited or viewed.
- The Edit CUT and PASTE command defaults have been added to the ISPF Configuration Table.
- The Edit CUT and PASTE default behaviors have been modified to use CUT REPLACE and PASTE KEEP.
- The BARRIER keyword has been added to the SELECT for Edit macros.
- A program called ISREMSPY that can be invoked from an Edit macro to display the current Edit data.
- The Edit macro commands CURSOR, LINENUM and DISPLAY_LINES can retrieve line numbers greater than 999999.

ISPF SCLM Component Changes

The ISPF SCLM component contains the following new functions and enhancements:

- Several enhancements to the Library Utility:
  - A member action to initiate Promotion on a member.
  - REFRESH command to update the member list contents.
  - HIER ON|OFF command to switch between full hierarchy view and single group view.
  - Edit action can create a new member when entered on the command line.
  - Ability to select deletion of accounting data or build map only.
- New FLMLRBLD macro to select automated rebuild for members with a specified language on promotion to listed groups.
- Improved edit models for SCLM services.
- VOL keyword on the FLMCPYLB and FLMSYSLB macros allowing reference to uncatalogued data sets.
- VIO keyword on the FLMALLOC macro to override the SCLM-calculated default unit of DASD or VIO for temporary data sets.
- Supplied parsers and translators are all loaded RMODE(31).

ISPF Client/Server Component Changes

The ISPF Client/Server Component enables a panel to be displayed unchanged (except for panels with graphic areas) at a workstation using the native display function of the operating system of the workstation. ISPF manuals call this "running in GUI mode."

There are no changes to the ISPF Client/Server for this release.

ISPF User Interface Considerations

Many changes have been made to the ISPF Version 4 user interface to conform to CUA guidelines. If you prefer to change the interface to look and act more like the Version 3 interface, you can do the following:

- Use the CUAATR command to change the screen colors
- Use the ISPF Settings panel to specify that the TAB or HOME keys position the cursor to the command line rather than to the first action bar item

- Set the command line to the top of the screen by deselecting ***Command line at bottom*** on the ISPF Settings panel
- Set the primary keys to F13–24 by selecting 2 for Primary range on the Tailor Function Key Definition Display panel
- Use the KEYLIST OFF command to turn keylists off
- Use the PSCOLOR command to change point-and-shoot fields to blue.
- Change the DFLTCOLR field in the PDF configuration table ISRCONFG to disable action bars and or edit highlighting

ISPF Migration Considerations

When migrating to OS/390 V2R8.0 or higher for the first time, you must convert your ISPF customization to the new format. Refer to the section entitled ***The ISPF Configuration Table*** in the *ISPF Planning and Customizing manual.*

When migrating from one version of ISPF to another, you must be sure to reassemble and re-link the SCLM project definition.

Note: If you are migrating to z/OS V1R2.0 from OS/390 V2R10.0, there are no migration actions necessary. If you are migrating to z/OS V1R2.0 from a prior release of OS/390, follow the migration actions for OS/390 V2R10.0.

ISPF Profiles

Major changes were made to the ISPF profiles for ISPF Version 4.2 and OS/390 Version 1 Release 1.0 ISPF. The profiles for ISPF Version 3 and the profiles for OS/390 ISPF are not compatible. If you are moving back and forth between an ISPF Version 3 system and OS/390 V1R1.0 or higher system, you must run with separate profiles. Profiles for OS/390 V1R1.0 and higher are compatible with each other.

Year 2000 Support for ISPF

ISPF is fully capable of using dates for the year 2000 and beyond. All of your existing applications should continue to run (some may need minor changes, as explained below) when the year 2000 comes. The base support for the year 2000 was added to OS/390 Version 1 Release 2.0, but the same level of support is available for ISPF Version 3.5, ISPF Version 4, and OS/390 Version 1 Release 1.0 as well. To get support for the earlier versions, be sure that your system has the correct APARs installed. All ISPF APARs that add or correct function relating to the year 2000 contain the YR2000 identifier in the APAR text. You should search for these APARs to ensure you have all the function available.

What function is included?

- ISPF Dialog variable ZSTDYEAR now correctly shows the year for dates past 1999. Earlier versions always showed the first 2 characters of the year as 19.
- A new ISPF dialog variable (ZJ4DATE) is available for Julian dates with a 4–digit year.
- An ISPF Configuration Table field enables PDF to interpret 2 character year dates as either a 19xx or 20xx date. The default value is 65. Any 2-character year date whose year is less than or equal to this value is considered a 20xx date, anything greater than this value is considered 19xx. To see what value has been set by the ISPF Configuration Table, use the new ZSWIND variable.
- New parameters in the LMMSTATS service (CREATED4 and MODDATE4) for specifying 4-character year dates. All existing parameters still exist and you can

continue to use them. If both the 2-character year date parameters (CREATED and MODDATE) and the 4-character year date parameters (CREATED4 and MODDATE4) are specified, the 2-character versions are used.

- Dialog variables ZLC4DATE and ZLM4DATE have been added.
  - You ***can*** set them before making an LMMREP or LMMADD call. Do this to specify a 4-character *created* or *last modified* date to set in the ISPF statistics.
  - They ***are*** set by LMMFIND, LMMLIST and LMMDISP to the current value of the created and last modified dates in the ISPF statistics.

What might need to change? Some minor changes to your existing ISPF dialogs might be necessary, especially in ISPF dialogs that use the Library Access Services to manipulate ISPF member statistics.

- For those services that accept both 4-character year dates and 2-character year dates, you can specify one or the other. If you specify both, the 2-character year date is used to avoid affecting existing dialogs. When the 2-character year date is used, the configuration table field mentioned above is used to determine whether the date should be interpreted as 19xx or 20xx.
- ISPF will not necessarily show 4-character dates in all circumstances but it will process them correctly. For example, a member list might only display 2-character year dates but will sort those dates in the proper order.
- SCLM stores dates past the year 1999 in a new internal format. If an accounting file contains dates in this new format, it cannot be processed by a system without year 2000 support. Accounting files without dates past 1999 can be processed with or without the year 2000 support.
- LMF has been removed from the ISPF product. For information about how to convert from LMF to SCLM refer to the ***ISPF Planning and Customizing*** manual.

What's in the z/OS V1R2.0 ISPF library?

You can order the ISPF books using the numbers provided below.

z/OS V1R2.0 ISPF

Title	Order Number
z/OS V1R2.0 ISPF Dialog Tag Language Guide and Reference	SC34-4824-01
z/OS V1R2.0 ISPF Planning and Customizing	GC34-4814-01
z/OS V1R2.0 ISPF User's Guide Volume I	SC34-4822-01
z/OS V1R2.0 ISPF User's Guide Volume II	SC34-4823-01
z/OS V1R2.0 ISPF Services Guide	SC34-4819-01
z/OS V1R2.0 ISPF Dialog Developer's Guide and Reference	SC34-4821-01
z/OS V1R2.0 ISPF Reference Summary	SC34-4816-01
z/OS V1R2.0 ISPF Edit and Edit Macros	SC34-4820-01
z/OS V1R1.0 ISPF Library Management Facility	SC34-4825-01
z/OS V1R2.0 ISPF Messages and Codes	SC34-4815-01
z/OS V1R2.0 ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide	SC34-4817–01
z/OS V1R2.0 ISPF Software Configuration and Library Manager Reference	SC34-4818-01
Entire library Bill of Forms	SBOF-8570

Elements and Features in z/OS

You can use the following table to see the relationship of a product you are familiar with and how it is referred to in z/OS Version 1 Release 2.0. z/OS V1R2.0 is made up of elements and features that contain function at or beyond the release level of the products listed in the following table. The table gives the name and level of each product on which a z/OS element or feature is based, identifies the z/OS name of the element or feature, and indicates whether it is part of the base or optional. For more compatibility information about z/OS elements see *z/OS Planning for Installation, GC28-1726*

Product Name and Level	**Name in z/OS**	**Base or Optional**
BookManager BUILD/MVS V1R3	BookManager BUILD	optional
BookManager READ/MVS V1R3	BookManager READ	base
MVS/Bulk Data Transfer V2	Bulk Data Transfer (BDT)	base
MVS/Bulk Data Transfer File-to-File V2	Bulk Data Transfer (BDT) File-to-File	optional
MVS/Bulk Data Transfer SNA NJE V2	Bulk Data Transfer (BDT) SNA NJE	optional
IBM OS/390 C/C++ V1R2	C/C++	optional
DFSMSdfp V1R3	DFSMSdfp	base
DFSMSdss	DFSMSdss	optional
DFSMShsm	DFSMShsm	optional
DFSMSrmm	DFSMSrmm	optional
DFSMS/MVS Network File System V1R3	DFSMS/MVS Network File System	base
DFSORT R13	DFSORT	optional
EREP MVS V3R5	EREP	base
FFST/MVS V1R2	FFST/MVS	base
GDDM/MVS V3R2 • GDDM-OS/2 LINK • GDDM-PCLK	GDDM	base
GDDM-PGF V2R1.3	GDDM-PGF	optional
GDDM-REXX/MVS V3R2	GDDM-REXX	optional
IBM High Level Assembler for MVS & VM & VSE V1R2	High Level Assembler	base
IBM High Level Assembler Toolkit	High Level Assembler Toolkit	optional
ICKDSF R16	ICKDSF	base
ISPF	ISPF	base
Language Environment for MVS & VM V1R5	Language Environment	base
Language Environment V1R5 Data Decryption	Language Environment Data Decryption	optional

Product Name and Level	Name in z/OS	Base or Optional
MVS/ESA SP V5R2.2		
BCP	BCP or MVS	base
ESCON Director Support	ESCON Director Support	base
Hardware Configuration Definition (HCD)	Hardware Configuration Definition (HCD)	base
JES2 V5R2.0	JES2	base optional
JES3 V5R2.1	JES3	base
LANRES/MVS V1R3.1	LANRES	base
IBM LAN Server for MVS V1R1	LAN Server	base
MICR/OCR Support	MICR/OCR Support	base
OS/390 UNIX System Services	OS/390 UNIX System Services	base
OS/390 UNIX Application Services	OS/390 UNIX Application Services	base
OS/390 UNIX DCE Base Services (OSF DCE level 1.1)	OS/390 UNIX DCE Base Services	base
OS/390 UNIX DCE Distributed File Services (DFS) (OSF DCE level 1.1)	OS/390 UNIX DCE Distributed File Services (DFS)	optional
OS/390 UNIX DCE User Data Privacy	OS/390 UNIX DCE User Data Privacy	optional
SOMobjects Application Development Environment (ADE) V1R1	SOMobjects Application Development Environment (ADE)	base
SOMobjects Runtime Library (RTL)	SOMobjects Runtime Library (RTL)	base
SOMobjects service classes	SOMobjects service classes	
Open Systems Adapter Support Facility (OSA/SF) R1	Open Systems Adapter Support Facility (OSA/SF)	base
MVS/ESA RMF V5R2	RMF	optional
OS/390 Security Server	Resource Access Control Facility (RACF) • DCE Security Server • OS/390 Firewall Technologies • Lightweight Directory Access Protocol (LDAP) Client and Server • Open Cryptographic Enhanced Plug-ins (OCEP)	optional
SDSF V1R6	SDSF	optional
SMP/E	SMP/E	base
	Softcopy Print	base
SystemView for MVS Base	SystemView for MVS Base	base
IBM TCP/IP V3R1 • TCP/IP CICS Sockets • TCP/IP IMS Sockets • TCP/IP Kerberos • TCP/IP Network Print Facility (NPF) • TCP/IP OS/390 Communications Service IP Applications • TCP/IP OS/2 Offload	TCP/IP • TCP/IP CICS Sockets • TCP/IP IMS Sockets • TCP/IP Kerberos • TCP/IP Network Print Facility (NPF) • TCP/IP OS/390 Communications Service IP Applications • TCP/IP OS/2 Offload	base • optional • optional • optional • optional • optional • optional
TIOC R1	TIOC	base
Time Sharing Option Extensions (TSO/E) V2R5	TSO/E	base

Product Name and Level	Name in z/OS	Base or Optional
VisualLift for MVS V1R1.1	• VisualLift Run-Time Environment (RTE) • VisualLift Application Development Environment (ADE)	• base • optional
VTAM V4R3 with the AnyNet feature	VTAM	base
3270 PC File Transfer Program V1R1.1	3270 PC File Transfer Program	base

Part 1. Introduction to ISPF

Chapter 1. The ISPF User Interface

The ISPF User Interface

ISPF provides an action bar-driven interface that exploits many of the usability features of Common User Access (CUA) interfaces. Refer to *Object-Oriented Interface Design: IBM Common User Access Guidelines* for additional information.

The panels look different than in Version 3: all screens are in mixed case, and most have action bars at the top. These action bars give you a new way to move around in the product as well as access to command nesting. Command nesting allows you to *suspend* an activity while you perform a new one rather than having to end a function to perform another function.

This chapter primarily explains the action bar-driven interface and the use of ISPF's graphical user interface (GUI).

Some Terms You Should Know

The following terms are used in this book:

action bar. The area at the top of an ISPF panel that contains choices that give you access to actions available on that panel. When you select an action bar choice, ISPF displays a *pull-down menu*.

pull-down menu. A list of numbered choices extending from the selection you made on the action bar. The action bar selection is highlighted; for example, Utilities in Figure 1 on page 5 appears highlighted on your screen. You can select an action either by typing in its number and pressing Enter or by selecting the action with your cursor. ISPF displays the requested panel. If your choice contains an *ellipsis* (...), ISPF displays a *pop-up window*. When you exit this panel or pop-up, ISPF closes the pull-down and returns you to the panel from which you made the initial action bar selection.

ellipsis. Three dots that follow a pull-down choice. When you select a choice that contains an ellipsis, ISPF displays a *pop-up* window.

pop-up window. A bordered temporary window that displays over another panel.

modal pop-up window. A type of window that requires you to interact with the panel in the pop-up before continuing. This includes cancelling the window or supplying information requested.

modeless pop-up window. A type of window that allows you to interact with the dialog that produced the pop-up before interacting with the pop-up itself.

point-and-shoot text. Text on a screen that is cursor-sensitive. See "Point-and-Shoot Text Fields" on page 8 for more information.

push button. A rectangle with text inside. Push buttons are used in windows for actions that occur immediately when the push button is selected (available only when you are running in GUI mode).

function key. In previous releases of ISPF, a programmed function (PF) key. *This is a change in terminology only.*

select. In conjunction with point-and-shoot text fields and action bar choices, this means moving the cursor to a field and simulating Enter.

mnemonics. Action bar choices can be defined with a underscored letter in the action bar choice text. In host mode you can access the action bar choice with the ACTIONS command and parameter 'x', where 'x' is the underscored letter in the action bar choice text. In GUI mode you can use a *hot key* to access a choice on the action bar; that is, you can press the ALT key in combination with the letter that is underscored in the action bar choice text.

How to Navigate in ISPF without Using Action Bars

If you use a non-programmable terminal to access z/OS V1R2.0 ISPF and you do not want to take advantage of the command nesting function, you can make selections the same way you always have: by typing in a selection number and pressing Enter.

How to Navigate in ISPF Using the Action Bar Interface

Most ISPF panels have action bars at the top; the choices appear on the screen in white by default. Many panels also have point-and-shoot text fields, which appear in turquoise by default. The panel shown in Figure 3 on page 6 has both.

Action Bars

Action bars give you another way to move through ISPF. If the cursor is located somewhere on the panel, there are several ways to move it to the action bar:

- Use the cursor movement keys to manually place the cursor on an action bar choice.
- Type ACTIONS on the command line and press Enter to move the cursor to the first action bar choice.
- Press F10 (Actions) or the Home key to move the cursor to the first action bar choice.

  If mnemonics are defined for action bar choices, you can:

  – In 3270 mode, on the command line, type ACTIONS and the mnemonic letter that corresponds to an underscored letter in the action bar choice text. This results in the display of the pull-down menu for that action bar choice.

  – In 3270 mode, on the command line enter the mnemonic letter that corresponds to an underscored letter in the action bar choice text, and press the function key assigned to the ACTIONS command. This results in the display of the pull-down menu for that action bar choice.

  – In GUI mode, you can use a *hot key* to access a choice on an action bar or on a pull-down menu; that is, you can press the ALT key in combination with the mnemonic letter that is underscored in the choice text to activate the text.

Use the tab key to move the cursor among the action bar choices. If you are running in GUI mode, use the right and left cursor keys.

Notes:

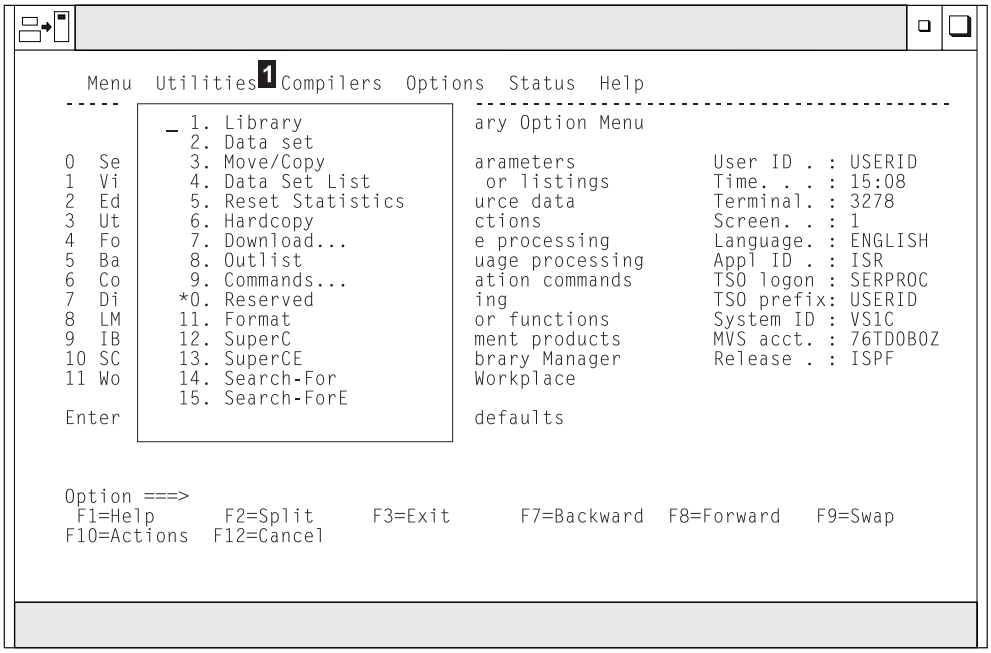
1. ISPF does not provide a mouse emulator program. This book uses *select* in conjunction with point-and-shoot text fields and action bar choices to mean moving the cursor to a field and simulating Enter.

   Note: Some users program their mouse emulators as follows:

   - Mouse button 1 – to position the cursor to the pointer and simulate Enter
   - Mouse button 2 – to simulate F12 (Cancel).

2. If you want the Home key to position the cursor at the first input field on an ISPF panel, type `SETTINGS` on any command line and press Enter to display the ISPF Settings panel. Deselect the **Tab to action bar choices** option.
3. If you are running in GUI mode, the Home key takes you to the beginning of the current field.

When you select one of the choices on the action bar, ISPF displays a pull-down menu. Figure 1 shows the pull-down menu displayed when you select Utilities on the ISPF Primary Option Menu action bar.



1 The selected action bar choice is highlighted.

Figure 1. Panel with an Action Bar Pull-Down Menu

To select a choice from the Utilities pull-down menu, type its number in the entry field (underlined) and press Enter or select the choice. To cancel a pull-down menu without making a selection, press F12 (Cancel). For example, if you select choice 9, ISPF displays the Command Table Utility pop-up, as shown in Figure 2 on page 6.

Note: If you entered a command on the command line prior to selecting an action bar choice, the command is processed, and the pull-down menu is never displayed. The CANCEL, END, and RETURN commands are exceptions. These three commands are not processed and the cursor is repositioned to the first input field in the panel body. If there is no input field, the cursor is repositioned under the action bar area. If you are running in GUI mode and select an action bar choice, any existing command on the command line is ignored.

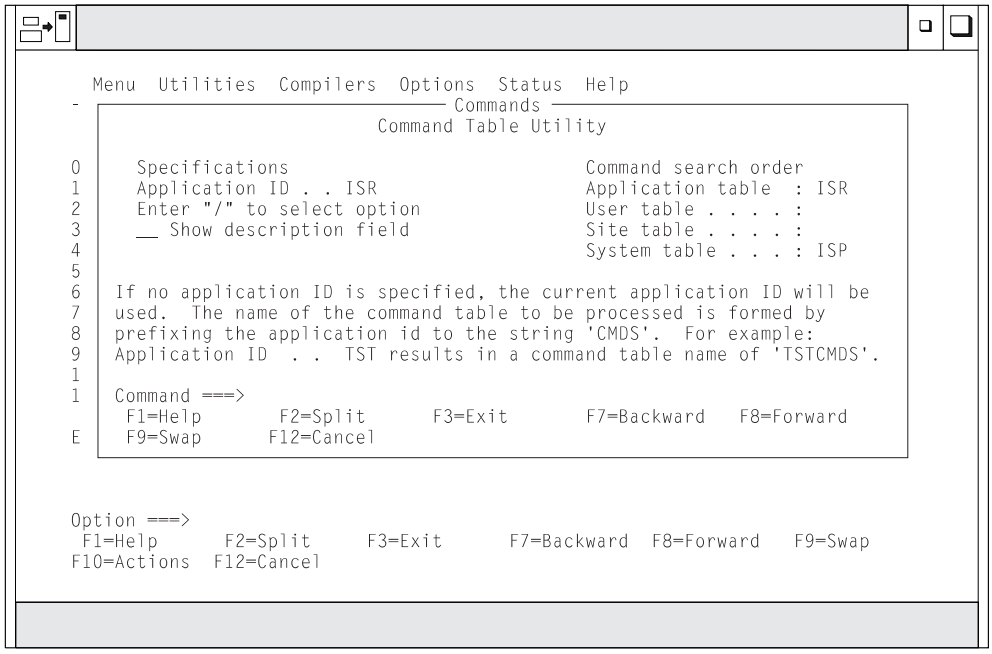
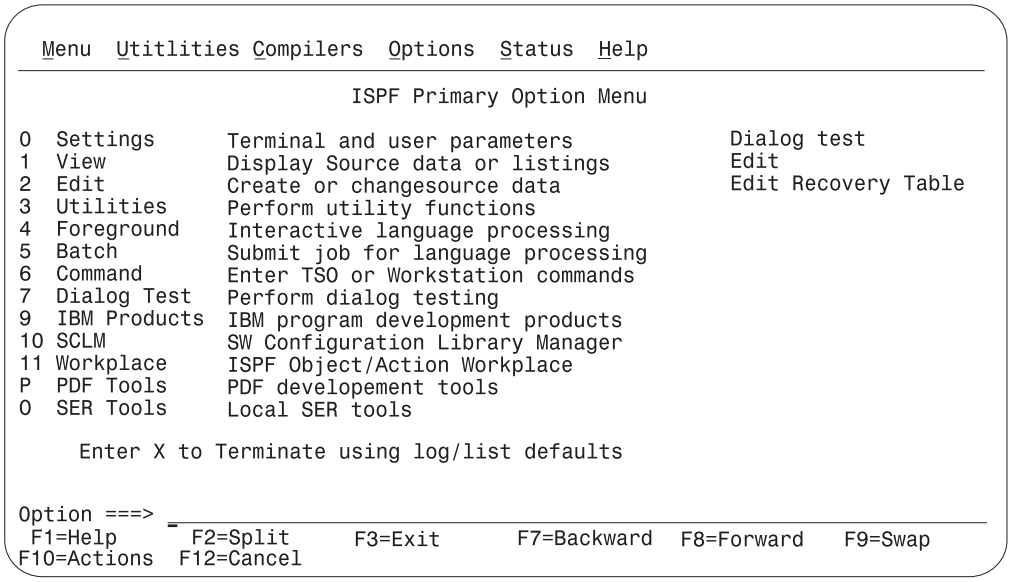


Figure 2. Pop-Up Selected from an Action Bar Pull-Down



1 Action bar. You can select any of the action bar choices and display a pull-down.

2 Options. The fields in this column are point-and-shoot text fields.

3 Dynamic status area. You can specify what you want to be displayed in this area.

Figure 3. Panel with an Action Bar and Point-and-Shoot Fields

Action Bar Choices

The action bar choices available vary from panel to panel, as do the choices available from their pull-downs. However, Menu and Utilities are basic action bar choices, and the choices on their pull-down menus are always the same.

Menu Action Bar Choice: The following choices are available from the Menu pull-down:

Settings	Displays the ISPF Settings panel
View	Displays the View Entry panel
Edit	Displays the Edit Entry panel
ISPF Command Shell	Displays the ISPF Command Shell panel
Dialog Test...	Displays the Dialog Test Primary Option panel
Other IBM Products...	Displays the Additional IBM Program Development Products panel
SCLM	Displays the SCLM Main Menu
ISPF Workplace	Displays the Workplace entry panel
Status Area...	Displays the ISPF Status panel
Exit	Exits ISPF.

Note: If a choice displays in blue (the default) with an asterisk as the first digit of the selection number (if you are running in GUI mode, the choice will be *grayed*), the choice is unavailable for one of the following reasons:

- Recursive entry is not permitted here
- The choice is the current state; for example, RefMode is currently set to Retrieve in Figure 4.

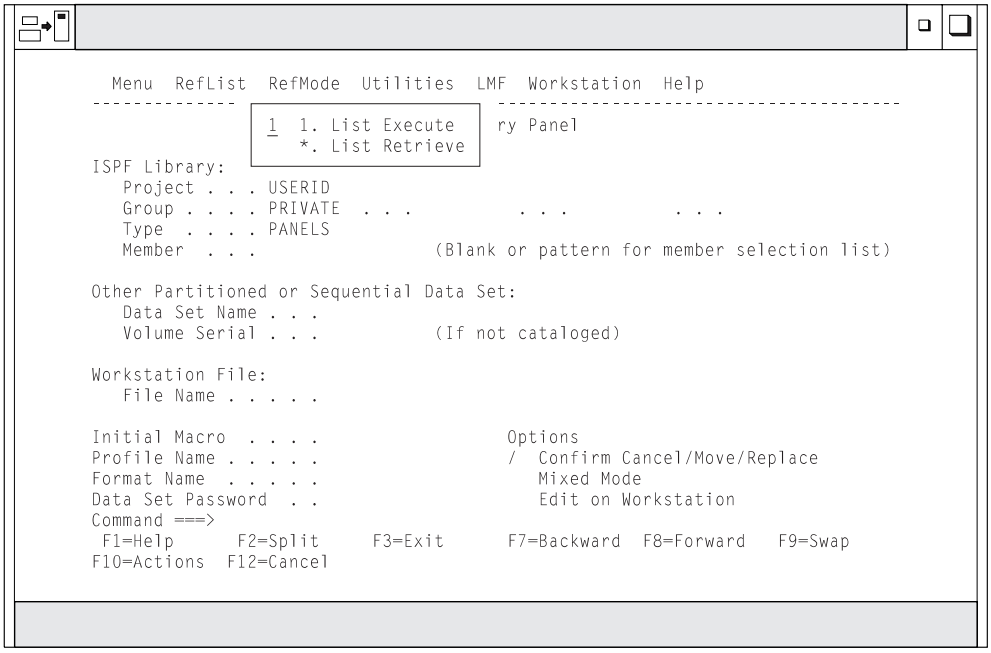


Figure 4. An Unavailable Choice on a Pull-Down

Utilities Action Bar Choice: The following choices are available from the Utilities pull-down:

Library	Displays the Library Utility panel
Data Set	Displays the Data Set Utility panel
Move/Copy	Displays the Move/Copy Utility panel

Data Set List	Displays the Data Set List Options panel
Reset Statistics	Displays the Reset ISPF Statistics panel
Hardcopy	Displays the Hardcopy Utility panel
Download...	Displays the panel that enables you to download workstation clients and other files from the host.
Outlist	Displays the Outlist Utility panel
Commands...	Displays the Command Table Utility panel
Reserved	Reserved for future use by ISPF; an unavailable choice
Format	Displays the Format Specification panel
SuperC	Displays the SuperC Utility panel
SuperCE	Displays the SuperCE Utility panel
Search-for	Displays the Search-For Utility panel.
Search-forE	Displays the Search-ForE Utility panel.

Point-and-Shoot Text Fields

Point-and-shoot text fields are cursor-sensitive; if you select a field, the action described in that field is performed. For example, if you select Option 0, Settings, in Figure 3 on page 6, ISPF displays the ISPF Settings panel.

Note: If you have entered a command on the command line, this command is processed before any point-and-shoot command unless you are running in GUI mode.

The cursor-sensitive portion of a field often extends past the field name. Until you are familiar with this new feature of ISPF, you might want to display these fields in reverse video (use the PSCOLOR command to set Highlight to REVERSE).

Note: You can use the Tab key to position the cursor to point-and-shoot fields by selecting the **Tab to point-and-shoot fields** option on the ISPF Settings panel (Option 0).

Function Keys

ISPF uses CUA-compliant definitions for function keys F1–F12 (except inside the Edit function). F13–F24 are the same as in ISPF Version 3. By default you see the CUA definitions because your **Primary range** field is set to 1 (Lower - 1 to 12).

To use non-CUA-compliant keys, select the **Tailor function key display** choice from the Function keys pull-down on the ISPF Settings (option 0) panel action bar. On the Tailor Function Key Definition Display panel, specify 2 (Upper - 13 to 24) in the **Primary range** field.

The following function keys help you navigate in ISPF:

F1 **Help**. Displays Help information. If you press F1 (and it is set to Help) after ISPF displays a short message, a long message displays in a pop-up window.

F2 **Split**. Divides the screen into two logical screens separated by a horizontal line or changes the location of the horizontal line.

Note: If you are running in GUI mode, each logical screen displays in a separate window.

F3 **Exit** (from a pull-down). Exits the panel underneath a pull-down.

F3 **End**. Ends the current function.

F7 **Backward**. Moves the screen up the scroll amount.

F8 **Forward**. Moves the screen down the scroll amount.

F9 **Swap**. Moves the cursor to where it was previously positioned on the other logical screen of a split-screen pair.

F10 **Actions**. Moves the cursor to the action bar. If you press F10 a second time, the cursor moves to the command line.

F12 **Cancel**. Issues the Cancel command. Use this command to remove a pull-down menu if you do not want to make a selection. F12 also moves the cursor from the action bar to the Option ==> field on the ISPF Primary Option Menu. See *ISPF Dialog Developer's Guide and Reference* for cursor-positioning rules.

F16 **Return**. Returns you to the ISPF Primary Option Menu or to the display from which you entered a nested dialog. RETURN is an ISPF system command.

Selection Fields

z/OS V1R2.0 ISPF uses the following CUA-compliant conventions for selection fields:

A single period (.)

Member lists that use a single period in the selection field recognize only a single selection. For example, within the Edit function you see this on your screen:

```
EDIT     USER1.PRIVATE.TEST               ROW 00001 of 00002
 Name         VV MM  Created   Changed  Size Init  Mod  ID
 . MEM1       01.00  94/05/12  94/07/22  40    0    0   USER1
 . MEM2       01.00  94/05/12  94/07/22  30    0    0   KEENE
```

You can select only one member to edit.

A single underscore (_)

Selection fields marked by a single underscore prompt you to use a slash (/) to select the choice. You may use any non-blank character. For example, the **Panel display CUA mode** field on the ISPF Settings panel has a single underscore for the selection field:

```
Options
  Enter "/" to select option
  _  Command line at bottom
  _  Panel display CUA mode
  _  Long message in pop-up
```

Note: If you are running in GUI mode, this type of selection field displays as a check box; that is, a square box with associated text that represents a choice. When you select a choice, a check mark (in OS/2) or an X (in Windows) appears in the check box to indicate that the choice is in effect. You can clear the check box by selecting the choice again.

An underscored field (____)

Member lists or text fields that use underscores in the selection field recognize multiple selections. For example, from the Display Data Set List Option panel, you may select multiple members for print, rename, delete, edit, browse, or view processing.

Command Nesting

Command nesting allows you to *suspend* an activity while you perform a new one rather than having to end a function to perform another function. For example, in

previous versions of ISPF, if you are editing a data set and want to allocate another data set, you type `=3.2` on the command line and press Enter. ISPF *ends* your edit session before taking you to the Data Set Utility panel. When you have allocated the data set and want to return to your edit session, you type `=2` and press Enter; ISPF returns you to the Edit Entry Panel. With z/OS V1R2.0 ISPF, from your edit session, select the Data set choice from the Utilities pull-down on the Edit panel action bar. ISPF suspends your edit session and displays the Data Set Utility panel. When you have allocated the new data set and end the function, z/OS V1R2.0 ISPF returns you directly to your edit session rather than to the Edit Entry Panel.

ISPF Client/Server -- The Workstation Agent Component

The ISPF Workstation Agent (WSA) component of z/OS V1R2.0 ISPF enables you to run ISPF on a programmable workstation and display the panels using the display function of your workstation operating system. Manuals in the ISPF library refer to this as *running in GUI mode*. The ISPF WSA is supported on the following platforms:

- OS/2
- Microsoft Windows
- AIX
- HP-UX
- Solaris

Why Use ISPF Client/Server?

Connecting to a workstation for data access has a direct impact on your installation's CPU processing time. One reason for using the ISPF Client/Server function is to offload CPU cycles from the host to a less expensive workstation. But even if that is not your goal, an added benefit is that your users can use the connection for *distributed editing*. This means that they can use their favorite editor to work with your data, whether that means using a host editor on host and workstation files, or using a workstation editor on the same files. By making the connection to the workstation, a user can edit workstation files on ISPF, or host files on his workstation.

But distributed edit does not stop there. The distributed edit function can be used in standard 3270 mode, or in ISPF GUI mode. This way, users that have spent years developing ISPF skills can use a familiar interface on a new workstation. Or, they can take advantage of the ISPF GUI mode. ISPF GUI was created to give you the ability to display any existing in-house or OEM ISPF applications in Graphical User Interface mode. Skilled ISPF application programmers who might lack workstation skills can write an ISPF GUI application using only the ISPF application development skills they already have.

Starting a GUI Session

Before you can start a GUI session, you must have completed the following tasks:

1. Successful installation and initialization of TCP/IP or APPC on your programmable workstation and the host.

   Notes:

   a. If the MVS TCP/IP Daemon name is not TCPIP, or if IP addresses for ISPF Client/Server connections are supplied in domain name format, the system programmer might need to update the ISPF Configuration table to support unique local TCP/IP configuration data set naming conventions. See *ISPF Planning and Customizing* for additional information.

b. The Workstation Agent does not attempt to initialize the communications protocol (either TCP/IP or APPC) after it receives a first initialization error. Therefore, the communications protocol for the workstation, either TCP/IP or APPC, *must* be started and active before you start the WSA.

2. Successful communications connection made between the workstation and the host.
3. Successful installation of the ISPF Workstation Agent component. See Appendix A. Installation Considerations for the Client/Server and the section of the *ISPF User's Guide Volume II* called ***Download Data Set to Workstation Utility (Option 3.7)*** for additional information.
4. Initialization of the WSA component.

You can start a GUI session in one of the following ways:

- Issue the ISPSTART command with the GUI parameter (see "Details for Dialog Developers Writing GUI Panels" on page 27 and refer to the *ISPF Dialog Developer's Guide and Reference* for additional information about the ISPSTART command. For example:

  ```
  ISPSTART GUI(IP:9.67.229.115)
  ```

  starts ISPF in GUI mode for the specified workstation.
- Select **Start GUI Session** on the Initiate GUI Session panel (option 0). See the **Workstation Connection** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for additional information.
- Issue the WSCON command from any command line to display the *Intiate Workstation Connection* panel. See "ISPF System Commands" on page 88 for more information.
- Call the WSCON service. Refer to the *ISPF Services Guide* for more information.

What GUI Mode Looks Like

This section describes changes that you see when you run ISPF in GUI mode.

Figures 5 through 12 show the differences between ISPF panels displayed on a 3278 emulator and a GUI display. Emulator screens shown are from an OS/2 Communications Manager 3278 emulator. The GUI display panels are shown as they would be displayed on a workstation running OS/2 WARP.

Figure 5 on page 12 shows the ISPF Primary Option Menu (with the copyright statement) displayed on a 3278 emulator.

```
 Menu  Utilities  Compilers  Options  Status  Help
                          ISPF Primary Option Menu

0  Settings      Terminal and user parameters           Dialog Test
1  View          Display source data or listings        Edit
2  Edit          Create or change source data           Edit Recovery Table
3  Utilities     Perform utility functions
4  Foreground    Interactive language processing
5  Batch         Submit job for language processing
6  Command       Enter TSO or Workstation commands
7  Dialog Test   Perform dialog testing
9  IBM Products  IBM program development products
                                                       r
 Licensed Materials - Property of IBM
 5694-A01 (C) Copyright IBM Corp. 1980, 2001.
 All rights reserved.
 US Government Users Restricted Rights -
 Use, duplication or disclosure restricted             s
 by GSA ADP Schedule Contract with IBM Corp.

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

Figure 5. ISPF Primary Option Menu Displayed on a 3278 Emulator

Figure 6 shows this panel displayed on a programmable workstation using ISPF WSA.

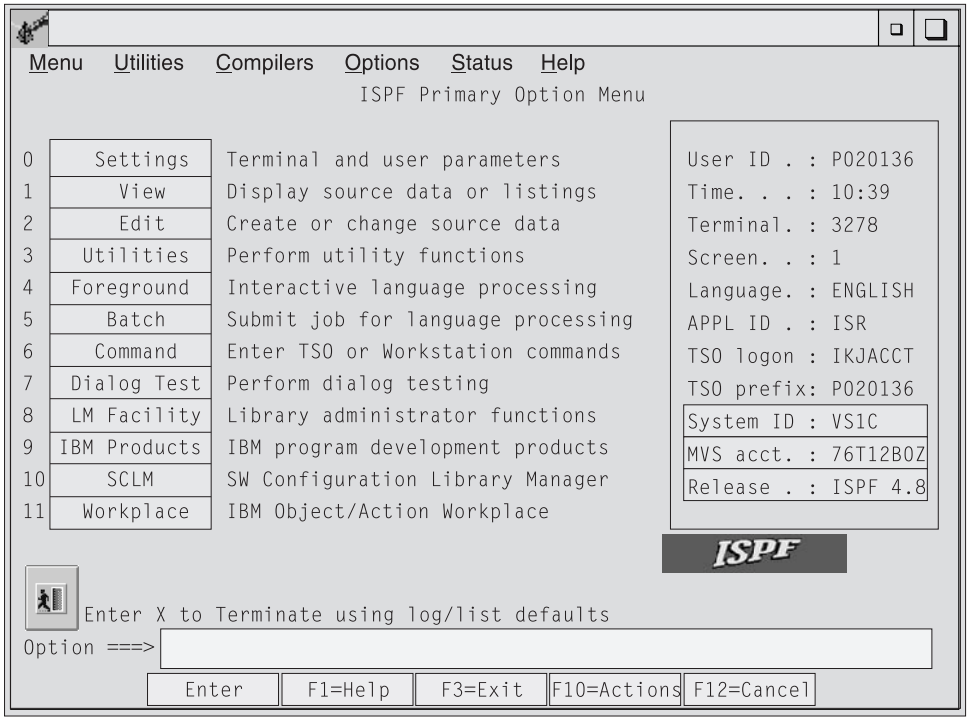


Figure 6. ISPF Primary Option Menu Displayed Using ISPF Workstation Agent

Action Bars and Pull-Down Menus

Action bars are responsive entities at the workstation; that is, pull-down menus display without issuing an interrupt to the host so you can browse the action bars quickly and easily. If you have entered a command on the command line, this command is ignored when you select a pull-down choice.

Title Bars

Various types of data can be displayed in the title bar. What data appears in the title bar depends on which of the following values ISPF finds data in first:

- The value defined in the application dialog variable ZWINTTL is used if the panel is displayed in a pop-up.
- The value defined in the application dialog variable ZAPPTTL.
- The value specified in the *title* variable on the TITLE parameter of the ISPSTART command.
- The value specified in the *title* variable on the TITLE parameter of the WSCON service.
- The value specified in the **GUI Title** field on the Initiate GUI Session panel available from option 0.
- Your user ID.

If PANELID is set to On, the panel ID displays in front of the window title (or your user ID, if no window title is defined) rather than on the first line of the panel.

Push Buttons

Function keys display as push buttons if FKA or PFSHOW is set to On. You can determine whether or not the F= prefix is shown on the push button by specifying PREFIX or NOPREFIX on the FKA command. Point-and-shoot output and text fields display as push buttons.

Note: If you entered a command on the command line, this command is ignored when you push a point-and-shoot push button.

Point-and-shoot input fields are cursor-selectable.

Cursor Movement

The up and down cursor keys move the cursor vertically through a group of input fields, point-and-shoot fields, and pull-down choices. Use the right and left cursor keys to move through the choices on an action bar.

You can group pushbuttons and checkbox fields together by using Cursor Groups. If you choose to use Cursor Groups in an application, the cursor up and cursor down keys move the focus through each of the fields within a group, and the TAB key moves the focus out of the group.

You can also set Radio buttons in an application. When the focus is within a radio button group, the up and down keys move the focus and the selection through the radio button choices.

Messages

Long and short messages are displayed as they are in non-GUI mode. If the message would appear in a pop-up window, it will be displayed in a message box. The message box will have CANCEL and HELP push buttons as well as the appropriate CUA-defined icon:

- An *i* in a circle (the international symbol for information) signifies an informational message.
- An exclamation point (!) signifies a warning message.
- A red circle with a diagonal line across it signifies an error message.

Note: You can force long messages into pop-ups using ISPF option 0.

Mnemonics

You can use a *hot key* to access a choice on an action bar or on a pull-down

menu; that is, you can press the ALT key in combination with the letter that is underscored in the choice to activate the choice.

Check Boxes
A selection field that is marked by a single underscore on the host displays as a check box (that is, a square box with associated text that represents a choice) if the attribute for the entry field has CKBOX(ON) and it is a valid check box field.

When you select a choice, some operating systems make a character (a ✓ in OS/2, an X in Windows) appear in the check box to indicate that the choice is in effect. You can clear the check box by selecting the choice again. Some operating systems (AIX and HP-UX) treat the check box as a pushbutton.

List Boxes
A control that enables you to display scrollable lists of choices in boxes. You can select a choice by tabbing to the desired choice and pressing Enter, or by positioning the mouse pointer on the desired choice. A single click selects the choice, a double click selects the choice and processes an Enter action. The vertical scroll bar of the list box enables you to scroll through the choices. The horizontal scroll bar enables you to scroll horizontally through the choices.

Drop-down Lists
A variation of a list box. A drop-down list only displays one item until you take action to display the other choices. You can select a choice by tabbing to the desired choice and pressing Enter, or by positioning the mouse pointer on the desired choice. A single click selects the choice, a double click selects the choice and processes an Enter action. The vertical scroll bar of the drop-down list enables you to scroll through the choices.

Combination Boxes
A combination box combines the functions of an entry field with a drop-down list. It has an entry field and contains a list of choices to select from to complete the entry field. The list of choices is hidden until you take action to display it.

You can also type text directly into the entry field. The text does not have to match one of the list choices.

The width of the combination box is the width of the input field.

Separator bars
You can provide visual distinction between two adjacent areas within a pull-down menu by coding a separator bar between the two areas.

Accelerators
You can use an accelerator to invoke an application-defined function from a pull-down menu. An accelerator is a key or combination of keys that you define.

Radio buttons
A radio button is a control that shows a fixed set of mutually exclusive choices, one of which is usually chosen.

To select a radio button, you can use your mouse pointing device to press the desired button. You can also use the TAB key to move the focus of your session to the currently selected radio button group, then use the cursor keys to move to the desired choice within the group. If the focus is on a radio button choice, the TAB key moves the focus to the next field or

group of fields following the first radio button choice. After one of the radio buttons is selected, you cannot blank it back out. Some applications handle this situation by using a radio button choice of *NONE*.

Unavailable Choices
Radio button, checkbox, and push button choices can display as unavailable choices. The unavailable choice is greyed out of the display and audible feedback occurs if you try to select it.

Function Keys
Function keys display as push buttons.

Panel Display CUA Mode
When you are running in GUI mode, the **Panel display CUA mode** option on the ISPF Settings panel is set to On.

Enter Key
An Enter key appears as a push button, by default, on all panels. You can control the display of this key from the GUI Settings panel (option 0).

Color, Intensity, and Highlighting
The Global GUI Color Change Utility is available (option 0) to allow you to map host colors to workstation colors. See the **Global GUI Color Change Utility** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for additional information. Character-level color, intensity, and highlighting are not supported; field-level intensity and highlighting are not supported.

Split Screen
When you split a screen, the new screen is displayed as a new physical window that is added to the window list. Each modeless window is single-threaded to MVS. SPLIT without parameters always acts as SPLIT NEW in GUI mode. Splitting a screen after the maximum number of screens is reached acts as a SWAP command. If you have only two split screens, SWAP simply toggles between the two. If you have more than two split screens, SWAP displays the Swap List.

PRINT Command
The PRINT, PRINT-HI, PRINTL, PRINTLHI, and PRINTG commands are not available in GUI mode.

WINDOW Command
The WINDOW command is not available in GUI mode.

WS Command
The WS command modelessly invokes the command you specify to provide a seamless interface between host and workstation applications. For example:

```
WS e.exe myfile
```

opens a modeless window and invokes the workstation editor on the file named *myfile*.

APL/TEXT Character Sets
The ZGE variable is set to Off when you are running in GUI mode. Any character defined with the GE(ON) attribute displays as a blank.

Graphic Areas
Graphic areas are not supported. When a GRINIT statement is encountered, you receive a message that panels with graphics cannot be displayed. You may choose to continue. When a panel with graphics is

encountered, a pop-up is displayed that allows you to specify that the panel be displayed on the host emulator session or on the workstation without the graphic.

Notes:

1. If you are in split screen mode, the graphic area panel cannot be displayed on the host session.
2. If you specified GUISCRD or GUISCRW values on the ISPSTART invocation that are different from the actual host screen size, GDDM cannot be initialized, and the GRINIT service ends with a return code of 20.

Input Fields

See "Sample Screens" on page 23 for more information on how input fields are displayed in GUI mode.

Note: Blank fields and trailing blanks are displayed as nulls in GUI mode in order to avoid the need to delete blank characters when replace mode is not available.

Fonts All GUI displays use the font that you specify in the **Font** option on the Workstation Agent window. If the GUI display of your panels does not appear as you expect, try using a monospace font.

Codepages and Character sets

Users invoking ISPF GUI from a terminal or emulator that does not support codepages (indicated in the terminal query response) should specify codepage and character set parameter values if values other than English are preferred when translating host data to workstation data.

Use the CODEPAGE and CHARSET parameters of the ISPSTART command, and the Host Codepage and Host Character Set fields on the ISPF Settings Initiate Workstation Connection panel, to set your chosen values. Refer to ***Extended Code Page Support*** in the *Dialog Developer's Guide and Reference* for codepage and character set values supported by ISPF.

Images

ISPF supports image files in the graphic interchange format (GIF) when running in GUI mode.

ISPF ships sample files in the sample library SISPSAMP. The panel ISR@PRIM uses three of the images (ISPFGIFL, ISPFGIFS, and ISPEXIT).

To use images, store the image files on the host in a partitioned data set and allocate this image data set to ddname ISPILIB before invoking ISPF. For more information about allocating this image library see "Allocating Optional Image ISPF Library" on page 169.

Ending a GUI Session

You can end a GUI session in one of the following ways:

- Terminate the ISPF session.
- Issue the WSDISCON command. See the WSDISCON entry of Table 3 on page 88 for more information.
- Invoke the WSDISCON dialog service. Refer to the *ISPF Services Guide* for more information.

After invocation of WSDISCON, the GUI screen continues to display back in the 3270 emulator session.

Switching Between GUI Mode and 3270 Mode

You can switch logical screens back and forth between GUI mode and 3270 mode by issuing the SWITCH 3270 or SWITCH GUI commands. See "The SWITCH Command" on page 63 for more information.

ISPF GUI Support of TSO Line Mode Output and Input

When running ISPF in GUI mode, users connected by TCP/IP on an OpenEdition MVS system, and all users connected by APPC, have the option to display all non-fullscreen TSO data in an ISPF/TSO GUI window. This window is scrollable and it contains an input field for entering required user responses. The data in the window can be selected and copied to a file of your choice.

This support enables you to minimize the emulator window from which you logged on without missing any messages that might appear in the emulator window.

Figure 7 shows the ISPF/TSO GUI window.

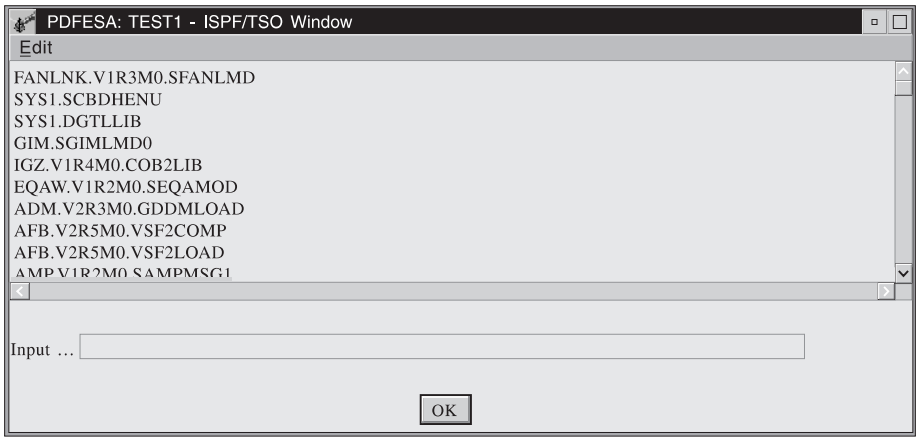


Figure 7. ISPF/TSO GUI Window

The ISPF/TSO window has one item on its action bar—**Edit**. Clicking on this option causes a pull-down menu to appear.

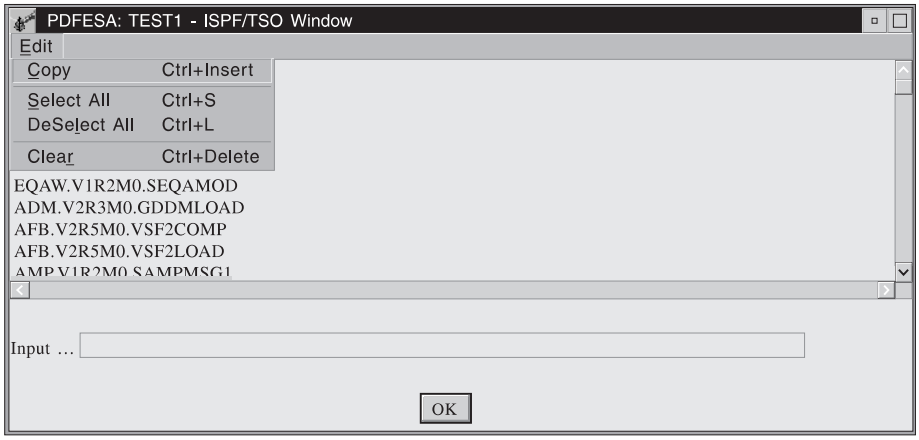


Figure 8. ISPF/TSO GUI Window with Pull-Down Menu

The options on the menu are as follows:

Copy Enables you to copy the lines of the window that you have marked. You designate a line to copy by clicking on it in the window.

Select All Mark all of the lines in the window for further action. The ISPF/TSO window is scrollable, and can contain 30,000 bytes of data in Windows, and 60,000 bytes of data in the other supported environments.

Deselect All Unmark all lines in the TSO window.

Clear Erase everything in the ISPF/TSO window.

The ISPF/TSO Window title contains the system name of the host system that you logged on to and your User ID. The system name is the **SYSNAME** value defined by your system programmer in the **IEASYSxx** member of **SYS1.PARMLIB**. For ISPF to display the line mode data in this ISPF/TSO Window, you must log on to TSO using a logon procedure that invokes the alternate entry point **IKJEFT1I**. In addition, for users running with TCP/IP communications, your system programmer must update the ISPF Configuration table keyword file, changing the USE_MVS_OPEN_EDITION_SOCKETS keyword value to YES, and rebuilding the configuration table load module. For more details on installation requirements for this TSO line mode support, refer your system programmer to the *ISPF Planning and Customizing* manual.

When you make a connection to the workstation to run ISPF in GUI mode, an additional connection is made for the ISPF/TSO Window. The window is created on your workstation when the first TSO message needs to be displayed. All messages are appended to this scrollable window until you end your ISPF session. At that time, the ISPF/TSO Window is closed and all TSO data from that point on is displayed on the 3270 emulator session.

Note: You cannot close the ISPF/TSO Window during the ISPF GUI session.

When running ISPF in GUI mode, if TSO input is required from the user, the ISPF/TSO Window is displayed in the foreground and the input field is enabled for you to enter your data.

Note: A maximum of 256 characters can be entered in this input field. When input is not being requested, the input field will be disabled.

The ISPF/TSO window has a maximum capacity of approximately 32000 bytes of data in a Windows environment, and 60000 bytes of data in all other ISPF-supported operating environments. After these limits are reached, a message is displayed informing you that the ISPF/TSO window will be cleared before appending the next line of data. This gives you a chance to use the data in the window. Using the options under the Edit menu item, you can select lines from the window, copy them, and paste them to any file of your choice. After you respond **OK** to the informational message, the window is cleared and any following TSO data is appended from the top of the window.

Restrictions for the TSO GUI Line Mode Support

Applications issuing TGETs must ensure that their TGET buffer is large enough for the user's response. The maximum allowable response is 256 characters. Normally, exceeding this limit results in a return code of 12 or 28 being returned. With the TSO GUI line mode support, a zero (0) return code is returned, and only the amount of data that fits in the supplied buffer is returned.

Users running in a Session Manager environment in the 3270 session will see any cross memory messages that appeared in the ISPF/TSO Window displayed again in the session manager screen after the ISPF GUI session ends.

Fullscreen TSO Data

Fullscreen TPUTs will continue to be displayed in the 3270 emulator session. Required input to satisfy a fullscreen TGET must be entered in the 3270 emulator session. For example, if the user from ISPF GUI mode invokes SDSF through its fullscreen TPUT interface (that is, the user enters **TSO SDSF** from an ISPF command line) the SDSF screens are displayed in the 3270 emulator screen.

However, if the user invokes SDSF through the ISPF interface ISPEXEC SELECT PANEL(ZSDSFOP2) NEWAPPL(ISF) then the SDSF screens are displayed in the ISPF GUI mode window just like any other ISPF panel.

Users running fullscreen applications like RMFMON, CSP, and OMVS from an ISPF GUI session with GUI TSO line mode support should first suspend the ISPF/TSO Window. This forces all fullscreen and line mode data to appear in the 3270 screen until the completion of the fullscreen application. The ISPF/TSO Window should be resumed at the completion of the fullscreen application processing.

You can query the current status of the ISPF/TSO Window using the **CONTROL TSOGUI QUERY** service. You can suspend and resume the window using one of the following methods:

1. ISPF Service

   ```
   CONTROL TSOGUI  QUERY|OFF|ON
   ```

   QUERY
   calls for the current status of the ISPF/TSO window. One of the following return codes is shown:

   - **0** — either the user is not running ISPF GUI with TSO line mode support or TSOGUI is off. All TSO input and output is directed to the 3270 session.

- **1** — all TSO line mode output displays in the ISPF/TSO window and line mode input must be entered into the ISPF/TSO window's input field.

OFF specifies that the ISPF/TSO window is suspended and all fullscreen and line mode data appears in the 3270 window until CONTROL TSOGUI ON is issued.

ON specifies that the ISPF/TSO window is resumed and all TSO line mode output and input is directed to the ISPF/TSO window.

Notes:

a. If you do not run ISPF GUI with TSO line mode support, issuing a CONTROL TSOGUI results in no operation being performed (NOP).
b. CONTROL TSOGUI defaults to ON during ISPF GUI session initialization.

EXAMPLE CLIST:

```
PROC 0
ISPEXEC CONTROL TSOGUI QUERY
SET &TSOGUIST= &LASTCC
IF (&TSOGUIST = 1) THEN +
  DO
   ISPEXEC CONTROL TSOGUI OFF
  END
ISPEXEC SELECT CMD (OMVS)
IF (&TSOGUIST = 1) THEN +
  DO
   ISPEXEC CONTROL TSOGUI ON
  END
```

2. ISPF command — TSOGUI

   The **TSOGUI** command is found in the ISPF command table as:

   ```
   SELECT PGM(ISPISM) PARM(TSOGUI,&ZPARM)
   ```

   From any ISPF command line you can enter:

   ```
   tsogui off
   ```

   or

   ```
   tsogui on
   ```

   to achieve your desired results.

Bi-directional Language Support

The ISPF workstation agent supports two bi-directional (bidi) languages —Arabic and Hebrew. This enablement of the workstation agent is supported on the following operating systems:

- OS/2
- Windows 95 or NT.

Shortcut Keys

With bi-directional versions of the OS/2 and Windows operating systems, support for special key sequences called shortcut, or "hot", keys, is provided to control bi-directional behavior within an application. For example, within an entry field on OS/2 you can press the keys Alt and Numlock to perform a field reverse.

The ISPF workstation agent supports these existing hot key combinations whenever possible. However, the workstation agent does provide some new or modified hot key combinations. The new or modified hot keys follow.

Note: The abbreviations "LtR" and "RtL" refer to the orientation of a field or application. LtR means *Left to Right* orientation. RtL means *Right to Left* orientation.

Screen Reverse (Alt+Enter): The term "screen reverse" is a remnant of ISPF's native 3270 support. Within an OS/2 or Windows application, this hot key combination reverses the application window, ***not*** the whole desktop screen. On both the OS/2 and the Windows operating systems, **Alt+Enter** is used to toggle the orientation of the screen. (Note that on an Arabic 102 keyboard, the Enter key is called **New Line**.

Auto Reverse (Alt+Numpad5): This key combination is used to automatically reverse the orientation of specific entry fields. On both OS/2 and Windows operating systems, use **Alt+Numpad5** to toggle between Auto Reverse **ON** and Auto Revers **OFF**.

When Auto Reverse is ON, all *numeric only* entry fields switch to LtR orientation, and alphanumeric entry fields switch to RtL orientation.

When Auto Reverse is OFF, all entry fields have the same orientation as the application screen.

Field Reverse (Ctrl+Left/RightShift): This combination works on **Windows only**, and has been modified from its normal Windows processing.

On the Windows operating system, the key combination **Ctrl+Left/RightShift** performs a field reverse. The effect of this is to change the keyboard language so that it matches the new typing direction. In a standard Windows application, this keyboard language change is global to the whole application window.

On a native 3270 display, the keyboard language change (resulting from a field reverse) effects only the target field, not the whole screen.

The ISPF workstation agent follows the behavior of a native 3270 display when this key combination is used on Windows.

Modifications to Arabic OS/2

These changes are applicable only to the Arabic OS/2 version of the ISPF Workstation Agent. The following menu items, because their effect is global to all open ISPF application windows, have been added to the System Menu.

Note: The abbreviations "LtR" and "RtL" refer to the orientation of a field or application. LtR means *Left to Right* orientation. RtL means *Right to Left* orientation.

Symmetric swapping: Select this menu item to toggle Symmetric swapping ON or OFF. The default condition is OFF. When symmetric swapping is ON, a check mark appears beside the menu item.

Symmetric swapping determines whether directional (symmetric) characters are automatically swapped within text segments when they are displayed RtL. For example:

```
Storage buffer: (1)Hello
Display:     olleH)1( -Visual, RtL text without symmetric swapping
Display:     olleH(1) -Visual, RtL text with symmetric swapping
```

Numeric swapping: Select this menu item to toggle Numeric swapping ON or OFF. The default condition is OFF. When numeric swapping is ON, a check mark appears beside the menu item.

Numeric swapping effects how host numerals are displayed on the workstation. If numeric swapping is ON, host numbers that are in Hindi are displayed in Arabic. Likewise, host numbers that are in Arabic are displayed in Hindi on the workstation.

Numeric display: Numeric display is a cascaded menu with three choices: **Arabic numerals**, **Hindi numerals**, and **Passthru numerals**. A check mark appears next to the menu item that has been chosen. The default choice is Arabic numerals.

Numeric display enables you to choose how numbers are displayed, either in Arabic form or Hindi form. Passthru is a special choice that prevents the operating system from performing any numeral conversions.

Note: If numeric *swapping* is ON, the numeric display is automatically set to Passthru numerals. Any previous setting is retained in the workstation agent INI file. When numeric swapping is turned OFF, the original numeric display setting is automatically restored.

Bi-directional File Transfer Limitations

Data files that contain all LtR text are not effected by the limitations described in this section. Neither is the workstation build function.

The ISPF Workstation Agent does not perform any transformations on the contents of files that are transferred to or from the host in text mode. The implications of this depend on how the files are edited on the workstation. If your host files contain Arabic or Hebrew text, consider the following points before editing them on the workstation:

- Host data is stored in Visual form. If you download a file to the workstation and edit it with an editor that expects the file contents to be in implicit form, the file can become corrupted.
- For Arabic OS/2, consider the following scenarios using the OS/2 System Editor:
  1. if the file editing is done in two steps (first download the file, then edit the file), and the editor is customized for Visual shaped data, then there is no problem.
  2. if the file download and editor invocation are done in one step (such as when using the Workstation Edit feature of ISPF), then there is a problem. The editor will, by default, work in Implicit base shapes. This is not compatible with the host.
- On Arabic Windows, the contents of a data file can become corrupted if the file is downloaded to the workstation then uploaded back to the host. This is because the host Arabic code page (420) supports shaped characters, while the Windows Arabic code page (1256) does not. Therefore, it is possible to lose shaping during the file transfer process.

RECOMMENDATION: If your host files contain Arabic or Hebrew text, *do not* edit them in GUI mode. Instead, edit these files on the host using the ISPF editor in its non-GUI mode.

Sample Screens

Note the following about GUI panels:

- Input fields appear as input boxes.
- If an input field wraps multiple lines on a 3278, the field scrolls horizontally in the GUI input box. Blank lines are displayed for those lines of the input field that would have wrapped on a 3278 display.
- For a panel not displayed in a pop-up window, only the amount of panel necessary to display the information is used; that is, ISPF does not generate useless blank space at the bottom of the panel. ISPF does display blank space coded into the panel.
- ISPF maintains the original window size on a pop-up window. If the panel exceeds the width or depth of the physical display, scroll bars are automatically added to allow you to view the hidden portions of the screen.

Figure 9 shows the ISPF Dialog Test Display Panel displayed on a 3278 emulator.

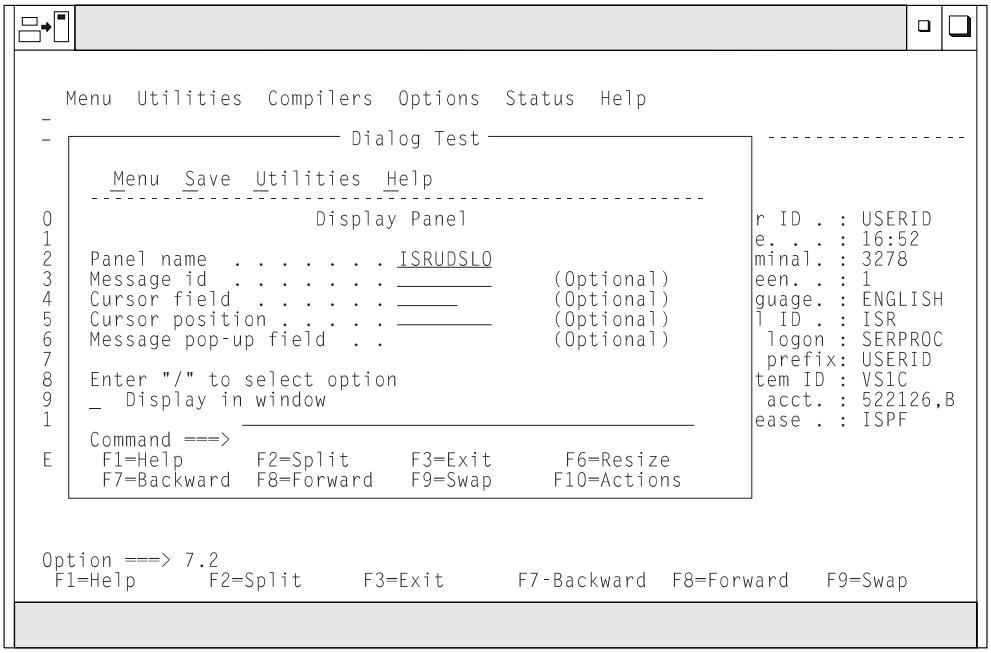


Figure 9. ISPF Dialog Test Display Panel Displayed on a 3278 Emulator

Figure 10 on page 24 shows this panel displayed on a programmable workstation using ISPF WSA.

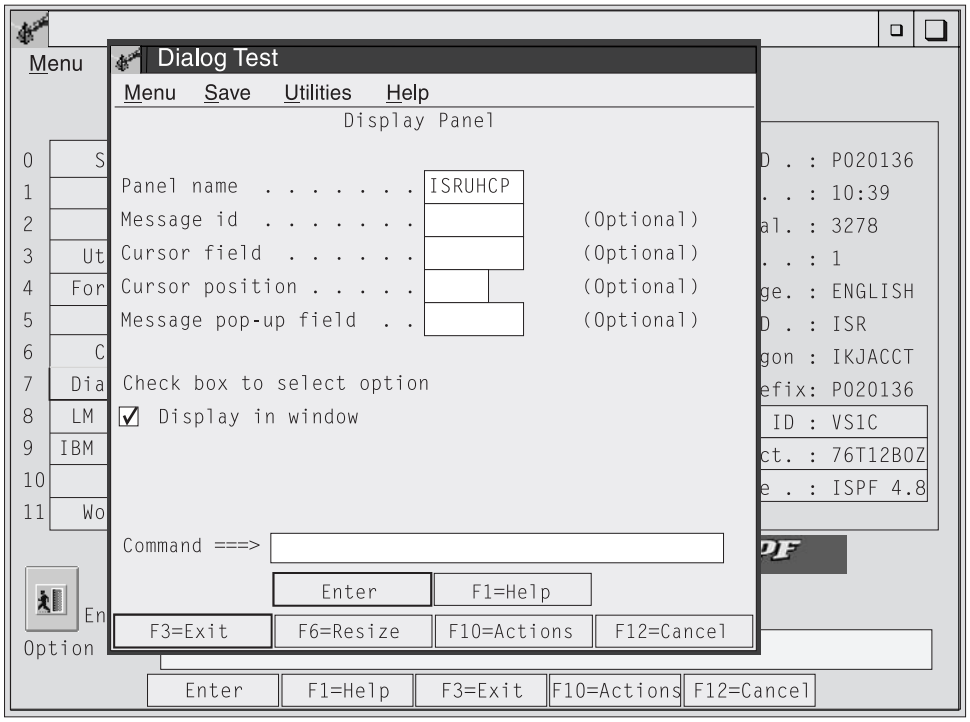


Figure 10. ISPF Dialog Test Display Panel Displayed Using ISPF Workstation Agent.

Figure 11 shows a pull-down with an unavailable choice (Session) displayed on a 3278 emulator.

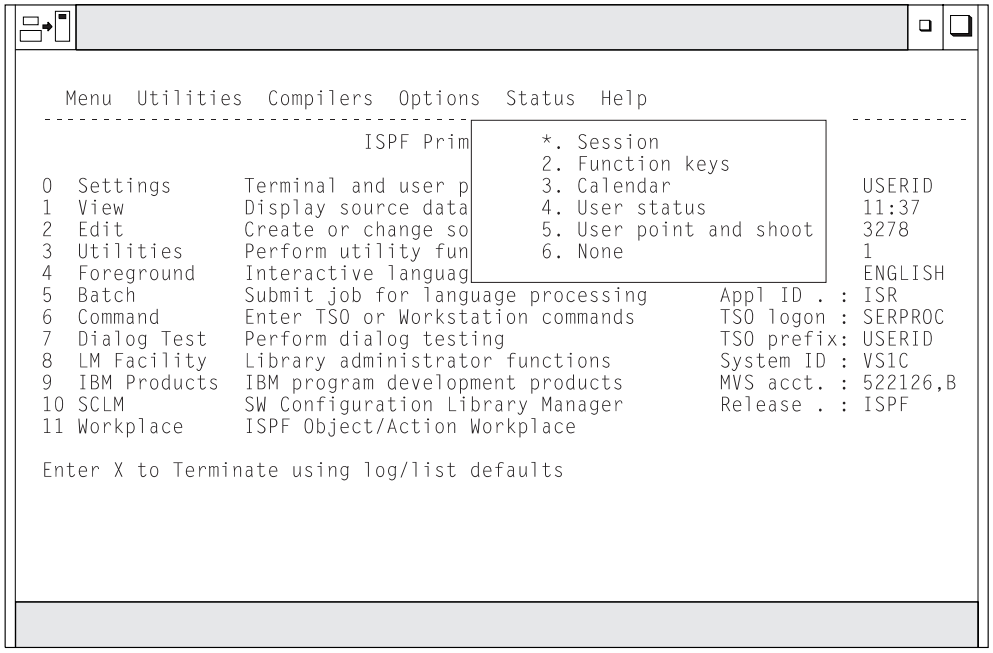


Figure 11. A Pull-Down Menu with an Unavailable Choice Displayed on a 3278 Emulator

Figure 12 on page 25 shows this panel displayed on a programmable workstation using ISPF WSA.

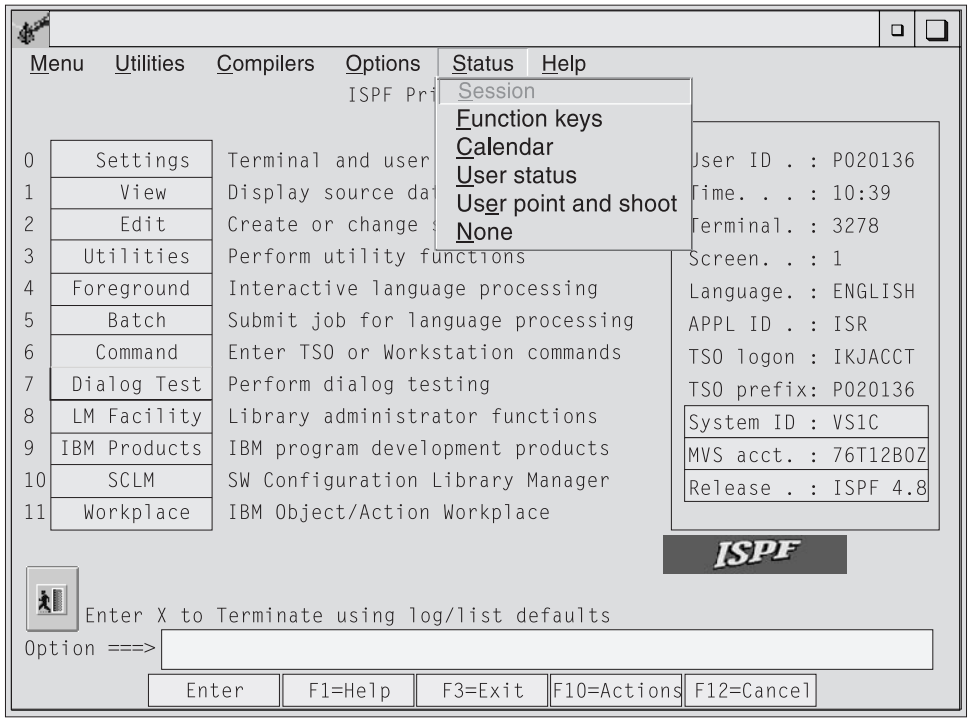


Figure 12. A Pull-Down Menu with an Unavailable Choice Displayed Using ISPF Workstation Agent

Some General GUI Restrictions

This section describes some restrictions that apply when you run z/OS V1R2.0 ISPF in GUI mode.

Cursor Placement

.CURSOR can be set only to an input or push button (point-and-shoot) field. If the application attempts to set the cursor to any other field, ISPF ignores the placement and uses the default cursor placement.

Character-Level Color, Intensity, and Highlighting

Character-level color, intensity, and highlighting are not supported when you are running in GUI mode.

Field-Level Intensity and Highlighting

Field-level intensity and highlighting are not supported when you are running in GUI mode.

Graphic Areas

Graphic areas are not supported. When a GRINIT statement is encountered, the user receives a message that panels with graphics cannot be displayed. The user may choose to continue. When a panel with graphics is encountered, a pop-up is displayed that enables you to specify that the panel be displayed on the host emulator session or on the workstation without the graphic.

Notes:

1. If you are in split-screen mode, the graphic area panel cannot be displayed on the host session.

2. If you specified GUISCRD or GUISCRW values on the ISPSTART invocation that are different from the actual host screen size, GDDM cannot be initialized, and the GRINIT service ends with a return code of 20.

Pop-Up Windows and Message Pop-Up Positioning

Dialog-specific pop-up positioning is not supported if you are running in GUI mode; that is, the POPLOC, ROW, and COLUMN parameters on the ADDPOP service are ignored. The MSGLOC parameter on the DISPLAY, SETMSG, and TBDISPL services is ignored.

SKIP Attribute

The panel attribute SKIP(ON) is ignored on the GUI display.

OUTLINE Attribute

The OUTLINE attribute is ignored on the GUI display.

3290 Partition Mode

You cannot invoke ISPF in GUI mode if you are configured to run ISPF in 3290 partition mode.

Closing the ISPF Client/Server Workstation Agent Component

If the ISPF WSA component window is closed while ISPF is running in GUI mode, ISPF issues a 989 abend on the host, unless the ***3270 mode after losing workstation connection*** field is selected on the GUI Settings panel. If this field is selected, the GUI session continues in your 3270 emulator session.

Special characters or symbols

The following describes techniques that can be used to display special characters or symbols while running ISPF in GUI mode. Although some are similar, the techniques are platform specific. In each case, the text refers to a number (sometimes in hexadecimal form) which is the codepoint of the character or symbol within the current (in use) codepage. Note that the codepoint for a symbol may have different values depending on the codepage being used. For example, the codepoint for the host not sign (¬) on OS/2 (using either codepage 437 or 850) is decimal 170. On Windows, the codepoint for the host not sign (using ANSI codepage 1252) is 172.

Windows

Use Microsoft's Character Map program to find the special character that you want to insert into your document or entry field. On Windows 95 and NT, the Character Map program is available from the Start menu. Choose Start, then Programs, then Accessories, and finally, Character Map.

After you have found the special character, select it by clicking with your mouse. Notice that the key sequence used to generate the character displays in the lower right portion of the window. You can use this key sequence in an entry field or GUI editor to display the character. Alternatively, you can use the Character Map program to copy the character to the clipboard, then paste it into your document.

When entering a character using its key sequence, remember that you must use the numeric keypad. For example, the key sequence for the host not sign (¬) is ALT+0172. To enter this key sequence, press and hold the ALT key, then press the numbers 0172 on the keypad, then release the ALT key. Note that the leading zero is important and must be specified in order to display the character

using the current (in use) character set. Without the leading zero, the OEM (or DOS) character set is used. Also note that within an entry field, some key sequences only work if NumLock is ON.

Refer to the Microsoft online help for the Character Map program for more information on using the program.

OS/2 OS/2 does not have a character map program like Microsoft Windows. However, you can still enter a key sequence to generate a special character or symbol. For example, the key sequence for the host not sign (¬) is ALT+170. To enter this key sequence, press and hold the ALT key, then press the numbers 170 on the keypad, then release the ALT key. Note that within an entry field, some key sequences only work if NumLock is ON.

UNIX (AIX)
On AIX you can enter a key sequence to generate a special character or symbol. The technique described for OS/2 works for AIX. Use ALT+172 instead of ALT+170.

UNIX (SOLARIS and HP)
SOLARIS and HP do not support this transformation technique. Instead, you can remap the keyboard. Remapping the keyboard works on all UNIX systems (AIX, SOLARIS, and HP). The following example shows how to remap **shift 6** to be the host not sign (¬).

1. execute **/usr/openwin/bin/xmodmap -pke > output.file**— this will dump the current mapping into the file named ***output.file***
2. edit ***output.file*** and replace **keycode 42 = 6 asciicircum** with **keycode 42 = 6 0XAC** (AC is the ascii equivalent of the not symbol)
3. execute **/usr/openwin/bin/xmodmap output.file**.

Once the keycode you are going to change is known, the change can also be made by executing the following:
/usr/openwin/bin/xmodmap -e 'keycode 42 = 6 0XAC' (the single quotes are required).

Note: The remap only effects new windows created after the command is run. Any existing windows will not honor the change. Also note that the change is only for the life of the current session. You might want to put the commands into some kind of startup that executes when you start a session. For more information about **xmodmap**, check the online documentaion on your system.

Details for Dialog Developers Writing GUI Panels

This section provides information that dialog developers need to write or adapt dialogs to run on a workstation. Refer to *ISPF Dialog Developer's Guide and Reference* for more information about dialogs.

How to Display an Application in GUI Mode

Use the GUI parameter on the ISPSTART command to invoke an application in GUI mode. See "Starting a GUI Session" on page 10, and the *ISPF Services Guide and Reference* for additional information.

ISPF Workstation Agent Component

GUI(LU:*address:tpname* | **IP:***address:port* | **FI:** | ,NOGUIDSP) **TITLE** (*title*)
FRAME(STD | FIX | DLG) **GUISCRW**(*screen-width*) **GUISCRD**(*screen-depth*) **CODEPAGE** (*codepage*) **CHARSET**(*character_set*)

where:

LU:*address:tpname*
The workstation's Advanced Program-to-Program Communication (APPC) network address and tpname.

An APPC address can be in fully-qualified LU name format or in symbolic destination name format. A fully-qualified LU name format consists of a network identifier and an LU name, separated by a period. For example, **USIBMNR.NRI98X00** is a fully-qualified LU name.

An APPC address in symbolic destination name format consists of a 1- to 8-character name such as **JSMITH**. The symbolic destination name must be defined as a *DESTNAME* and the corresponding fully-qualified LU name must be defined as the associated *PARTNER_LU* in the APPC/MVS side information.

If specified, the tpname is used to construct the names of the two transaction programs required to support an ISPF Client/Server connection. The ISPF Client/Server function appends different single alphabetic characters to the supplied name to form the actual names of the two APPC transaction programs.

If the tpname is used, the same tpname must be specified from the Options action bar choice on the WSA.

IP:*address:port*
The workstation's Internet Protocol (IP) address and TCP/IP port.

A TCP/IP address can be in dotted decimal format or in domain name format. Dotted decimal format is a sequence of decimal numbers separated by periods, for example, **9.87.654.321**.

A TCP/IP address in domain name format consists of one or more domain qualifiers separated by periods. The minimum specification for addresses within the same domain is a TCP/IP host name, for example, **jsmith**. The fully-qualified domain name for *jsmith* is formed by appending the appropriate subdomain name and root domain name to *jsmith*, such as **jsmith.raleigh.ibm.com**. To use domain naming, a domain name server must be active and providing domain name resolution for domain names within your TCP/IP network. The domain name server address is determined by the value of the **NSINTERADDR** statement in the TCP/IP configuration data set. ISPF must be able to locate the TCP/IP configuration data set as described in the **TCP/IP Requirements Detail** section of the ***Configuring Communications for the ISPF Client/Server*** appendix of the *ISPF User's Guide Volume I*.

FI: Specifies that you want to search a file allocated to DD ISPDTPRF for the user's network protocol and workstation address to be used when initiating a workstation connection or GUI display. For example, the system programmer could maintain a file containing all of the user's workstation addresses so all users would be able to use the same logon procedure or startup CLIST to run ISPF GUI.

The file itself can be sequential or a member of a PDS. It can be fixed block (FB) or variable blocked (VB). Each line of the file should be formatted as follows:

```
userid WORKSTAT protocol_id:network_address
```

Where:

userid
user's TSO userid

protocol_id
network protocol identifier: ***ip*** for TCP/IP or ***lu*** for APPC.

network_address
workstation address

For example, `KRAUSS WORKSTAT ip:7.30.200.94` might be one line of your file.

EXAMPLES OF ISPSTART SYNTAX USING FI: OPTION

To specify that you want ISPF to search the file allocated to ISPDTPRF DD for your network address when connecting to the workstation from ISPSTART, and to run ISPF in GUI mode, enter `ISPSTART GUI(FI:)`. To specify that you want to search the file, but to run ISPF in 3270 mode, enter `ISPSTART GUI(FI:,NOGUIDSP)`. For more information about this, refer to the *Dialog Developer's Guide and Reference.*

NOGUIDSP
Specifies that you want to make a connection to the workstation, but DO NOT want ISPF to display in GUI mode. For more information about this, refer to the *Dialog Developer's Guide and Reference.*

TITLE(*title*
The default value for the title bar variable. This value has a maximum length of 255 characters and can be truncated without notice to the user at display time.

FRAME(STD | FIX | DLG)
Specifies that the first window frame displayed be a standard (STD), fixed (FIX), or dialog (DLG) window frame.

Note: Pop-up panels are always displayed in dialog window frames.

GUISCRW(*screen-width***)**
Enables you to specify a screen width different than that of the emulator or real device from which you enter the ISPSTART command. If you do not specify GUISCRD, the depth is that of the emulator or real device.

If GUISCRW is different than the emulator or real device and GUI initialization fails, ISPF does not initialize. Dialogs started with dimensions other than those of the emulator or real device that use the GRINIT service cannot display GDDM screens.

GUISCRD(*screen-depth***)**
Enables you to specify a screen depth different than that of the emulator or real device from which you enter the ISPSTART command. If you do not specify GUISCRW, the width is that of the emulator or real device.

If GUISCRD is different than the emulator or real device and GUI initialization fails, ISPF does not initialize. Dialogs started with dimensions other than those of the emulator or real device that use the GRINIT service cannot display GDDM screens.

The variable ZGUI is set to the workstation address (in character format) if ISPSTART is issued with the GUI parameter; ZGUI is set to blank if ISPSTART is issued without the GUI parameter.

Note: Users can force the application into GUI mode using the ISPF Settings panel (option 0). The display address specified on this panel is saved across ISPF sessions.

CODEPAGE(*codepage***) CHARSET(***character_set***)**
When running in GUI mode or connecting to the workstation, these values are used as the host codepage and character set in translating data from the host to the workstation, regardless of the values returned from the terminal query response.

Other Considerations

Action Bars and Pull-Down Menus
Action bars are responsive entities at the workstation and do not require an interrupt to the host to display a pull-down menu. All)ABCINIT sections run prior to sending the panel to the workstation. The)ABCPROC section runs after the pull-down has been selected at the workstation.

Title Bars
Various types of data can appear in the title bar, depending on the following values. The first of the following values for which ISPF finds data is displayed in the title bar:

- The value defined in the application dialog variable ZWINTTL is used if the panel is displayed in a pop-up
- The value defined in the application dialog variable ZAPPTTL.
- The value specified in the *title* variable on the TITLE parameter of the ISPSTART command.
- The value specified in the **GUI Title** field on the Initiate GUI Session panel available in option 0.
- The value specified in the *title* variable on the TITLE parameter of the WSCON service.
- Your user ID.

ZWINTTL and the *title* variable on ISPSTART have a maximum length of 255 characters and can be truncated without notice to the user at display time if they do not fit on the panel.

Messages
If a short or long message would appear in a pop-up window in non-GUI mode, it is displayed in a message box in GUI mode. The message box includes the appropriate icon as defined by CUA guidelines:

- .TYPE=NOTIFY produces a question mark (?).
- .TYPE=WARNING produces an exclamation point (!).
- .TYPE=ACTION or .TYPE=CRITICAL produces a red circle with a diagonal line across it.

Closing a Window
If a user closes a window (that is, selects Close from the system menu), ISPF returns the CANCEL, END, EXIT, or RETURN command to the dialog, as specified on the GUI Settings panel (option 0).

Function Keys
You cannot give a function key the default focus.

Check Boxes
Check boxes are supported at the workstation if CKBOX(ON) is set for a one-character entry field that is followed by an output field.

List Boxes
List boxes are supported at the workstation. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information.

Drop-down Lists
Drop-down lists are supported at the workstation. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information.

Group Boxes
Group boxes are supported at the workstation. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information.

Combination Boxes
Combination boxes are supported at the workstation. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information.

Unavailable Choices
Unavailable choices for check boxes, radio buttons, and push buttons are supported at the workstation. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information.

Mnemonics
Mnemonics are supported at the workstation in action bar and pull-down menu choices using the MNEM keyword on the ABC and PDC statements.

Separator Bars
Separator bars group logically related choices in pull-down menus. Use the PDC keyword PDSEP to display separator bars.

Accelerators
Accelerators are assigned to menu choices so those choices can be initiated quickly, even when the menu that the choice appears on is not currently displayed. Use the PDC keyword ACC to implement accelerators.

Radio Buttons
Radio buttons provide a way to select mutually exclusive choices. Use the)ATTR keyword RADIO to set radio buttons.

Enter Key
An Enter key push button appears, by default, on all panels. You can change the text on the push button using the ZENTKTXT variable.

Note: If a dialog sets ZENTKTXT to blanks, the Enter push button is not displayed even if you select the Display Enter Key option on the GUI Settings panel available from option 0.

APL/TEXT Character Sets
The ZGE variable is set to Off when you are running in GUI mode. Any character defined with GE(ON) displays as a blank.

Cursor Placement
.CURSOR can be set only to an input or push button (point-and-shoot) field. If the application attempts to set the cursor to any other field, ISPF ignores the placement and uses the default cursor placement. The up and down cursor keys move vertically through a group of input fields, point-and-shoot fields, and pull-down choices.

Images
ISPF supports image files in the graphic interchange format (GIF) when running in GUI mode.

ISPF ships sample files in the sample library SISPSAMP. The panel ISR@PRIM uses three of the images (ISPFGIFL, ISPFGIFS, and ISPEXIT).

To use images, store the image files on the host in a partitioned data set and allocate this image data set to ddname ISPILIB before invoking ISPF. For more information about allocating this image library see "Allocating Optional Image ISPF Library" on page 169.

The ISPF Workstation Tool Integration Program

ISPF enables you to more fully utilize your desktop workstation's potential by giving you the ability to edit host data on the workstation, and workstation data on the host. ISPF calls this function *distributed editing*.

The ISPF Workstation Tool Integration dialog, or tool integrator, is a workstation customization tool that enables any workstation application to use data from an MVS host system. After setting up the tool integrator, your workstation-installed applications can interact with the ISPF View and Edit functions and services. Data flow goes both ways with the tool integrator connection. You can work with workstation files on the host or with host files on the workstation.

Overall, the tool integrator enables MVS-based users to use workstation applications and then share data through MVS. MVS, its file system, and RACF, provide data integrity, security, and distribution capabilities beyond the workstation. Sharing data this way can reduce your installation's dependence on complex LANs, off-load CPU cycles to your workstation, and take advantage of workstation tools.

Starting the Tool Integration Dialog

Before you can use the tool integrator, you must have completed the following tasks:

1. Successful installation and initialization of TCP/IP or APPC on your programmable workstation and the host.

   Notes:

   a. If the MVS TCP/IP Daemon name is not TCPIP, or if IP addresses for ISPF Client/Server connections are supplied in domain name format, the system programmer might need to update the ISPF Configuration table to support unique local TCP/IP configuration data set naming conventions. See *ISPF Planning and Customizing* for additional information.

   b. The Workstation Agent does not attempt to initialize the communications protocol (either TCP/IP or APPC) after it receives a first initialization error. Therefore, the communications protocol for the workstation, either TCP/IP or APPC, *must* be started and active before you start the WSA.

2. Successful communications connection made between the workstation and the host.
3. Successful installation of the ISPF Workstation Agent component. See Appendix A. Installation Considerations for the Client/Server and the section of the *ISPF User's Guide Volume II* called ***Download Data Set to Workstation Utility (Option 3.7)*** for additional information.
4. Initialization of the WSA component.

You can run the tool integrator in either GUI mode or the standard ISPF mode. For information about how to run ISPF in GUI mode, see the **Workstation Connection**

section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II*. The examples used in this section of the book show the panels in GUI mode.

Note: You cannot issue a WSDISCON command or service call while running the workstation tool integration dialog.

There are several ways to start the Workstation Tool Integration dialog and display the first entry panel. You can choose the **Workstation** choice on the action bars found on the Edit or View panels, then choose *Workstation Tool Integration*. Or, you can type **INT** on any ISPF command or action line and press Enter. The panel that appears is shown in Figure 13.

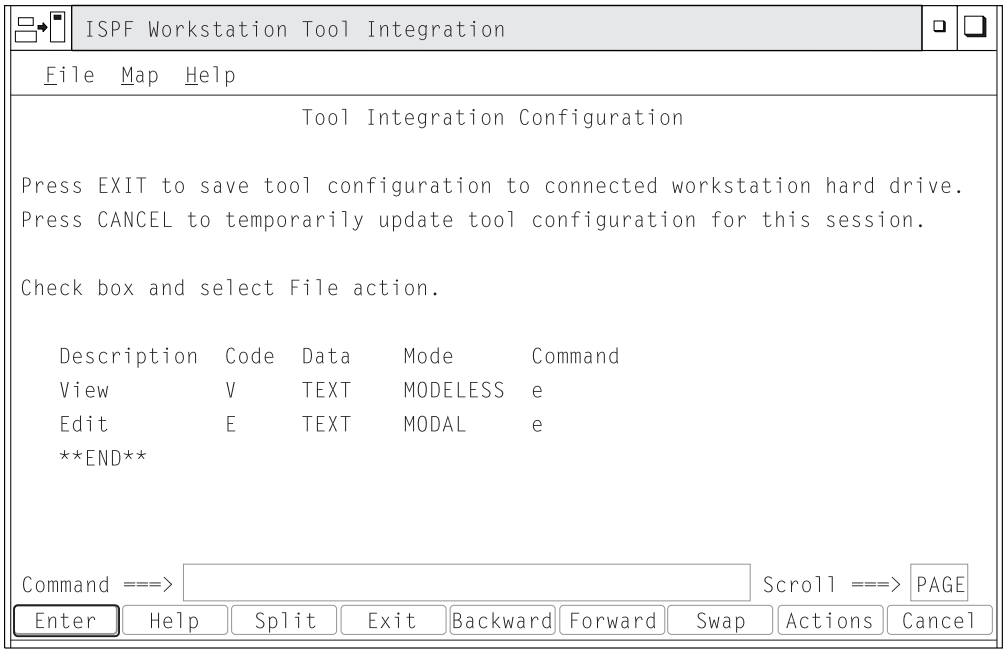


Figure 13. ISPF Workstation Tool Integration window

The information contained on this panel is defined as follows:

Description Text explaining the ISPF action.

Code The ISPF action code assigned to this workstation application.

Data The format used for the object file during download and upload to the host.

Mode The processing mode for this action.

Command The workstation invocation command string or name.

You use this panel to begin to define how the ISPF actions interact with the workstation applications, or just to look at the actions you have already defined. To view the configuration of an action, put a check mark in the check box next to the word *View* on the panel.

The Tool Integrator Action Bar Choices

The following action bar choices are available on the Workstation Tool Integration panel:

File

Edit... Calls the edit function to update the parameters of a workstation action.

View... Displays the parameters of a particular action.

Cancel changes Exit edit without updating the workstation action.

Exit and Save Exit edit and save changes made to the workstation action parameters.

Map

Extension
Displays the Extension Mapping window and starts the extension mapping dialog. See “Extension Mapping” on page 35 for more information.

Updating an Action

To update the configuration of an action, place a check mark in the checkbox next to *Edit* on the Tool Integration Configuration panel.

After you choose to edit an action, the screen shown in Figure 14 appears.

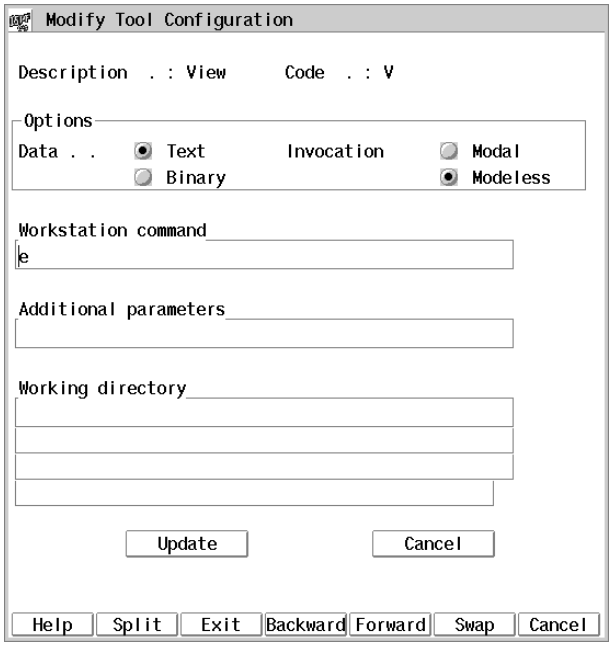


Figure 14. Modify Tool Configuration Window

The fields found on this panel are as follows:

Description
Text explaining the ISPF action.

Code The ISPF action code assigned to this workstation application.

Data The format used for the object file during download and upload to the host. Use Binary for data that you want to leave unchanged on the host, for example, image files such as bitmaps.

Invocation
The processing mode for this action. Use Modal to ensure that an action is completed before your application moves on to another task. For example, if you want to upload and replace an object on the host, use Modal for the mode. Modeless means that the action can continue while your application performs other tasks, effectively running in parallel with the application.

Workstation Command
The workstation invocation command string or name. This can be the actual command name of the workstation application, or a workstation path and command name.

Additional Parameters
Additional parameters for the workstation command you defined in the **Workstation Command** field.

Working Directory
The directory on the workstation that is defined to the host as the location that is used when Workstation Edit is used. This directory is used to temporarily store the host data when Edit or View on the workstation has been selected, or to save workstation data when a workstation file has been selected to be edited on the workstation. This directory is also used when the workstation command is started.

If you do not specify a directory location, then the workstation edit working directory defaults to the ISPF WSA's defined working directory. The default is the directory where the ISPF WSA is started (in OS/2 and Windows), or the user's home directory (in AIX, HP-UX, and SUN), *not* the location of the DLL or executable.

Extension Mapping

The **Map** choice on the tool integrator action bar enables you to specify a default character set (3 characters or less) to correlate host data set types to a workstation file extension. The host data set type is the last qualifier of the data set. These are frequently longer than 3 characters, while most workstation file extensions are 3 or less. The Extension Mapping dialog panel shown in Figure 15 on page 36 shows how certain host data set types map to their workstation file extension counterparts.

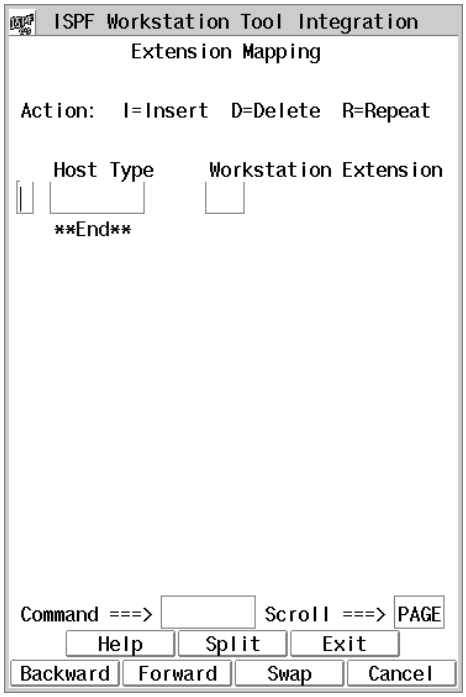


Figure 15. Extension Mapping Dialog Window

To map a host to workstation pair, perform one of the following:

1. in the **Action** field, type R to repeat a previous mapping, I to insert a new mapping, or D to delete a current mapping
2. in the **Host Type** field, type the last qualifier of the host data set to be mapped
3. in the **Workstation Extension** field, type the extension name to be used for the temporary workstation file created
4. press Enter.

If you attempt to use a host data set on a workstation and you have not mapped the data set type to one on your workstation, the first 3 characters of the data set type are used by default.

Chapter 2. Overview of ISPF

ISPF helps programmers develop interactive applications called *dialogs*.

Dialogs are interactive applications because ISPF uses them to communicate with terminal users through a series of panels while the users do application development tasks.

ISPF panels:
- Provide access to ISPF functions (menus)
- Request information (data entry panels)
- Provide information (scrollable data displays).

The following sections describe the functions provided by the ISPF licensed program.

ISPF Functions

ISPF can be used in the following ways:
- Managers can use ISPF Edit, SCRIPT/VS, and the Hardcopy utility or another print utility provided by their installation to prepare memos.
- Data processing administrators and system programmers can use ISPF to:
  - Monitor and control program libraries
  - Communicate with MVS through TSO commands, CLISTs, or REXX EXECs.
- Programmers can use ISPF to develop a batch, interactive, or any other type of program and its documentation.
- Terminal users can call dialogs that use Dialog Manager (DM) component and Program Development Facility (PDF) component dialog services to do the work of the application.

The View, Browse, and Edit functions, a wide range of utilities, foreground and batch compilers, program library control, and other facilities are available to help you develop ISPF dialogs.

View, Browse, Edit, Edit Macros, and Models

The View, Browse, and Edit functions allow you to look at the contents of a dialog. While editing a dialog, you can change it by adding or deleting lines, typing over the existing source code, or copying lines from another dialog to the one being edited.

To enhance the existing Edit function, you can write edit macros. Edit macros allow you to combine several often-used functions so that you do not have to call each function separately. You can write initial edit macros that are automatically run when the Edit option is selected. Other uses for edit macros include:
- Overriding Edit commands
- Calling DM and PDF component dialog services
- Accessing cursor position and data location.

Also, ISPF provides online models that you can insert into the dialog. A *model* is an example of a service call, panel format, table format, or message that contains the

proper syntax and all the available parameters for the programming language being used. Since these models are online, they can be called directly into the member being edited.

Refer to *ISPF Edit and Edit Macros* for more information.

Dialog Services

The PDF component provides View, Browse, Edit, and library access services that can be combined in a dialog with any of the ISPF services. The library access services carry out functions involving members of a programming library. These functions include adding, finding, and deleting members, and displaying member lists.

The PDF component includes a separate edit model of each service call for each programming language ISPF supports: CLIST, COBOL, EXEC, FORTRAN, PL/I, Pascal, C, and REXX. See *ISPF Services Guide* for complete information about the PDF component dialog services.

Note: For information about library access services that apply to the Software Configuration and Library Manager (SCLM), refer to the *ISPF Software Configuration and Library Manager (SCLM) Developer's and Project Manager's Guide*

Utilities

ISPF provides a wide range of utilities. Utilities enable you to:

- Display and print library and data set member lists
- Reset statistics for ISPF library members
- Define commands to be used with specific dialogs
- Compare data sets and search for strings of data
- Move, copy, and print library and data set members.

Dialog Test, Foreground, and Batch

ISPF provides special facilities for testing dialogs. The Dialog Test option allows testing of individual dialog elements and complete dialogs.

After a program has been developed, you can either assemble it or compile and link it using either the Foreground or Batch option. The Foreground option allows you to watch the program being compiled. The Batch option frees the terminal, which helps when you compile a long program. If errors occur, you can use the debugging capabilities of the Dialog Test facility to correct them.

TSO Commands, CLISTs, and REXX EXECs

While using ISPF, you can call TSO commands, CLISTs, or REXX EXECs by:

- Prefixing a command with TSO and entering it from any command line
- Selecting the Command option and entering the TSO command, CLIST, or REXX EXEC on the panel
- Entering a TSO command, CLIST, or REXX EXEC in the **Line Command** field on a data set list display or a member list display.

Software Configuration and Library Manager

Another way you can maintain different levels or versions of a library member is to use the Software Configuration and Library Manager (SCLM). SCLM is a software tool that helps you develop complex software applications. Throughout

the development cycle, SCLM automatically controls, maintains, and tracks all of the software components of the application. And you can lock the version being edited in a private library and then promote it. Refer to *ISPF Software Configuration and Library Manager (SCLM) Developer's and Project Manager's Guide* for more information.

Other IBM Program Development Products

You might want to call another IBM program development product while using ISPF. The IBM Products option allows you to call the Cross System Product/Application Development, Cross System Product/Application Execution, Information/Management, COBOL Structuring Facility foreground dialog, and COBOL Structuring Facility background dialog licensed programs without leaving ISPF.

Online Tutorial

Learning to use ISPF is made easier by the following:

- Online help and tutorial facilities (available while using ISPF).

For example, if you need help filling in the data requested by an ISPF utility, you can use the tutorial to help you understand the data entry requirements for that utility.

Recursive ISPF

ISPF is not supported in recursive mode except in those options that specifically document recursive support.

Supported Data Types

ISPF supports partitioned (PDS), sequential (SEQ), and partitioned extended (PDSE) data sets. These data sets can be used in any of the ISPF options, such as Edit and Browse. ISPF does *not* support the following:

- VSAM data sets (except for creating, obtaining data set information, and deleting)

  Note: VSAM data sets ***are*** supported for Edit, Browse, and View ***if*** the ISPF Configuration table has been customized to enable the support.
- Record format variable block spanned (VBS) data sets
- Direct access data sets
- Tape data sets
- Multivolume data sets for the ISRLEMX program, SCLM, and File Tailoring
- Generation data group (GDG) base data sets
- Deletion of data sets allocated with an esoteric device type
- Member aliases
- PDSEs as the ISPF control file
- Unmovable data sets under the ISPF Move/Copy utility (option 3.3) or using the LMMOVE or LMCOPY service
- Data sets allocated with the BUFNO parameter (ISPF handles its own buffering)
- Browse for packed multivolume data sets.

ISPF requires exclusive enqueues on data sets for many of its functions. If a data set is allocated as SHAREd to a user and then is operated on by one of these

functions, the allocation will be converted to OLD by MVS dynamic allocation. This allocation of OLD may remain after ISPF frees its enqueue. This is a restriction of the MVS operating system.

Member Name Conventions

ISPF allows you to create member names that follow this ISPF naming convention:

1st character
Alphabetic or special (@,#,$)

2nd–8th characters
Alphabetic, numeric, or special.

All member names created within ISPF are converted to uppercase. If you create members outside of ISPF that do not meet these conventions, they are displayed in ISPF member lists and can be selected from those lists. These member names can also be specified for the Browse service with the exception of member names containing lowercase alphabetics. (ISPF converts the member name to uppercase before searching for the member and therefore cannot process a lowercase member.) Member names not meeting the ISPF naming convention are not supported for the other ISPF services.

Note: ISPF does not support using option M (member list) from a data set list and then selecting E to edit a member whose name contains lowercase letters. ISPF uses the Edit service in this case, and its services do not support lowercase member names.

Interacting with ISPF

As an interactive dialog, ISPF communicates with you through panels and messages. Ordinarily, the first panel you see when you enter the ISPF command is the ISPF Primary Option Menu, shown in Figure 16 on page 42. Panels display data, selection lists, and data-entry fields, such as a data set name or an ISPF command.

ISPF responds interactively to the information or command you enter by displaying another panel, displaying a message, or carrying out a command. For more information about how panels and messages are displayed, see “Understanding ISPF Panels” on page 53.

One helpful aspect of your interaction with ISPF is the online tutorial. If you need information about using the online tutorial, see the *ISPF Dialog Developer’s Guide and Reference*

Starting ISPF

To start ISPF, log on to TSO. When the READY prompt appears, type `ISPF` or `PDF` and press Enter. If your installation has established an alias for ISPF, such as SPF, enter that instead.

The ISPF and PDF commands are aliases of ISPF module ISRPCP. When you run ISRPCP (or one of its aliases), ISPF is invoked with the command `ISPSTART PANEL(ISR@PRIM) NEWAPPL(ISR)`. You may specify any of the parameters valid for ISPSTART when invoking ISPF, PDF, or ISRPCP. However, if you specify parameters on the ISPF or PDF commands, only those parameters are passed to ISPSTART. For example, if you specify:

```
ISPF NEWAPPL(ABC)
```

ISPF will be invoked with the command:

```
ISPSTART NEWAPPL(ABC).
```

Notice that ISPF does not pass PANEL(ISR@PRIM), part of its normal default string, to ISPSTART. Because the ISPSTART command generated does not contain a PANEL(...), PGM(...) or CMD(...) keyword, the invocation in the preceding example causes you to remain at the Ready prompt. You must ensure that you specify all the parameters you need to run your application in the correct environment.

For information about ISPSTART, refer to the *ISPF Dialog Developer's Guide*.

Option Selection

You can select an ISPF option three ways:

- Select a choice from one of the pull-downs on the action bar. See Figure 1 on page 5 for more information.
- Select one of the point-and-shoot fields. See "Point-and-Shoot Text Fields" on page 8 for more information.
- Type an option number on the Option line and press Enter.

Many options have a secondary list of options. To bypass the second menu, type two selections, separating them with a period, on the ISPF Primary Option Menu. For example, entering 3.1 on the ISPF Primary Option Menu is the same as entering 3 on the ISPF Primary Option Menu and 1 on the Utility Selection Panel.

An even faster way to select an option is to bypass both the ISPF Primary Option Menu and the secondary menus. To do this, include your options in the ISPF (or alias) command. For example:
ISPF 2 To go directly to the Edit option.
ISPF 3.1
To go directly to the Library utility (3.1).

ISPF Primary Options

Figure 16 on page 42 shows the first panel, the ISPF Primary Option Menu.

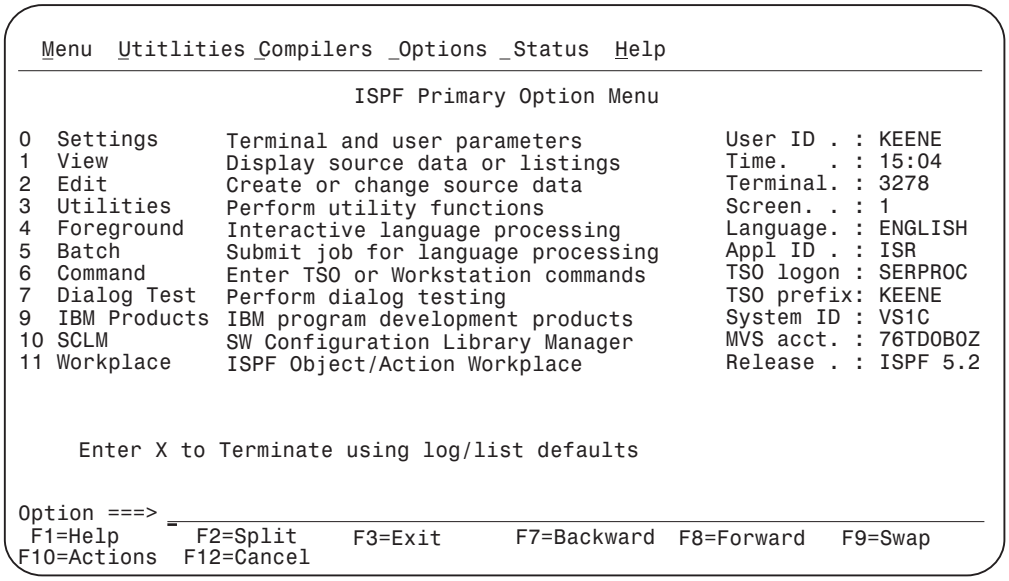


Figure 16. ISPF Primary Option Menu (ISR@PRIM)

Primary Option Menu Action Bar Choices

The Primary Option Menu action bar choices function as follows:

Menu See "Menu Action Bar Choice" on page 7 for information on the Menu pull-down.

Utilities

See "Utilities Action Bar Choice" on page 7 for information on the Utilities pull-down.

Compilers

The Compilers pull-down offers you the following choices:

1 **Foreground Compilers** Displays the Foreground Selection Panel.

2 **Background Compilers** Displays the Batch Selection Panel.

3 **ISPPREP Panel Utility...** Displays the Preprocessed Panel Utility panel.

4 **DTL Compiler...** Displays the ISPF Dialog Tag Language Conversion Utility panel.

Options

The Options pull-down offers you the following choices:

1 **General Settings** Displays the ISPF Settings panel.

2 **CUA Attributes...** Displays the CUA Attribute Change Utility panel.

3 **Keylists...** Displays the Keylist Utility panel.

4 **Point-and-Shoot...** Displays the CUA Attribute Change Utility panel, positioned on the Point-and-Shoot panel element.

5 **Colors...** Displays the Global Color Change Utility panel.

6 **Dialog Test appl ID...** Displays the Dialog Test Application ID pop-up to allow you to change the application ID for Dialog test so

that you can look at variables in the application profile for an application that runs under a different application ID than the one under which ISPF was started (by default, ISR).

Status The Status pull-down offers you the following choices:

1 **Session**
2 **Function keys**
3 **Calendar**
4 **User status**
5 **User point and shoot**
6 **None**

Note: The current setting will be shown as an unavailable choice; that is, it will display in blue (the default) with an asterisk as the first digit of the selection number (if you are running in GUI mode, the choice will be *grayed*).

See "Status Area on the Primary Option Menu" on page 45 for more information on using these choices to tailor the status area.

Help The Help pull-down offers you the following choices:

1 General
2 Settings
3 View
4 Edit
5 Utilities
6 Foreground
7 Batch
8 Command
9 Dialog Test
10 LM Facility
11 IBM Products
12 SCLM
13 Workplace
14 Exit
15 Status Area
16 About...
17 Changes for this Release
18 Tutorial
19 Appendices
20 Index

Option Descriptions

When you select one of these options, ISPF displays the selected panel. These options are described in detail in the *ISPF User's Guide Volume II*. Brief descriptions follow:

Option Description

0 **Settings** displays and changes selected ISPF parameters, such as terminal characteristics and function keys. See the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for more information.

1 **View** displays data (you cannot change it) using the View or Browse function. Use View or Browse to look at large data sets, such as compiler listings. You can scroll the data up, down, left, or right. If you are using Browse, a FIND command, entered on the command line, allows you to search the data and find a character string. If you are using View, you can

use all the commands and macros available to you in the Edit function. See the ***View (Option 1)*** chapter of the *ISPF User's Guide Volume II* for more information.

2 **Edit** allows you to create or change source data, such as program code and documentation using the ISPF full-screen editor. You can scroll the data up, down, left, or right. You can change the data by using Edit *line commands*, which are entered directly on a line number, and *primary commands*, which are entered on the command line. See the ***Edit (Option 2)*** chapter of the *ISPF User's Guide Volume II*, and refer to *ISPF Edit and Edit Macros* for more information.

3 **Utilities** perform library and data set maintenance tasks, such as moving or copying library or data set members, displaying or printing data set names and volume table of contents (VTOC) information, comparing data sets, and searching for strings of data. See the ***Utilities (Option 3)*** chapter of the *ISPF User's Guide Volume II* for more information.

4 **Foreground** calls IBM language processing programs in the foreground. See the ***Foreground (Option 4)*** chapter of the *ISPF User's Guide Volume II* for more information.

5 **Batch** calls IBM language processing programs as batch jobs. ISPF generates Job Control Language (JCL) based on information you enter and submits the job for processing. See the ***Batch (Option 5)*** chapter of the *ISPF User's Guide Volume II* for more information.

6 **Command** calls TSO commands, CLISTs, or REXX EXECs under ISPF. See the ***Command (Option 6)*** chapter of the *ISPF User's Guide Volume II* for more information.

7 **Dialog Test** tests individual ISPF dialog components, such as panels, messages, and dialog functions (programs, commands, menus). See the ***Dialog Test (Option 7)*** chapter of the *ISPF User's Guide Volume II* for more information.

8 **LM Facility** controls and tracks the movement of members into controlled libraries and throughout library hierarchies. See the ***Library Management Facility (Option 8)*** chapter of the *ISPF User's Guide Volume II*, and refer to *ISPF Library Management Facility* for more information.

9 **IBM Products** allows you to select other installed IBM program development products on your system. Products supported are:

- Cross System Product/Application Development (CSP/AD)
- Cross System Product/Application Execution (CSP/AE)
- Information/System (INFO/SYS)
- COBOL Structuring Facility foreground dialog (COBOL/SF-F)
- COBOL Structuring Facility background dialog (COBOL/SF-B)
- Screen Definition Facility II (SDF II) licensed program
- Screen Definition Facility II-P (SDF II-P) licensed program.

See the ***IBM Products (Option 9)*** chapter of the *ISPF User's Guide Volume II* for more information.

10 **SCLM** controls, maintains, and tracks all of the software components of an application. See the ***SCLM (Option 10)*** chapter of the *ISPF User's Guide Volume II*, and refer to *ISPF Software Configuration and Library Manager (SCLM) Developer's and Project Manager's Guide* for more information.

11 **Workplace** gives you access to the ISPF Workplace, which combines many

of the ISPF functions onto one object-action panel. See the ***ISPF Object/Action Workplace (Option 11)*** chapter of the *ISPF User's Guide Volume II* for more information.

X **EXIT** leaves ISPF using the log and list defaults. You can change these defaults from the Log/List pull-down on the ISPF Settings panel action bar.

Status Area on the Primary Option Menu

The status area on the ISPF Primary Option Menu is a 21-column dynamic area that is composed of a 12-character description field, one attribute byte, and an 8-character field to display the value of the selected variable. The status area is limited to eleven description fields and their values. It can be manipulated from two places:

- The Status pull-down on the ISPF Primary Option Menu action bar. Use this pull-down to specify *what* you want to display in the status area. See "Status Pull-Down" for additional information and examples.
- The Status Area... choice from the Menu pull-down that is available on most action bars throughout ISPF. Use this facility to define the *contents* of the status area. See "Status Area Choice" on page 52 for additional information and examples.

The first five logical screens each have their own status view. For each screen after that, the view defaults to the setting of the first screen.

Status Pull-Down

When you select one of the choices in the Status pull-down on the ISPF Primary Option Menu action bar (shown in Figure 17), you specify what you want to display in the status area.

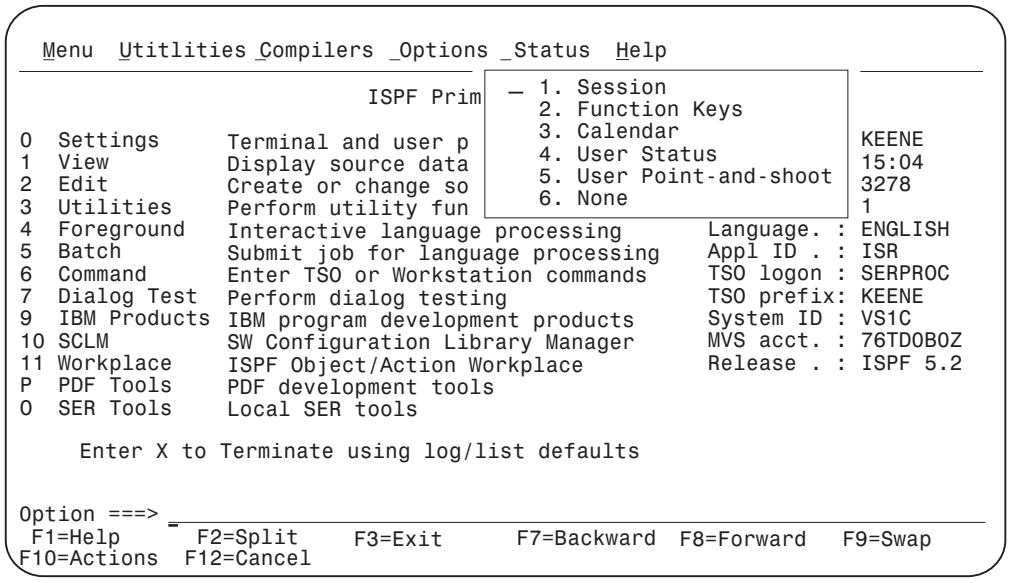


Figure 17. Status Pull-Down on the ISPF Primary Option Menu (ISR@PRIM)

Note: The current setting is shown as an unavailable choice; that is, displays in blue (the default) with an asterisk as the first digit of the selection number. If you are running in GUI mode the choice is *grayed*.

Status Area on the Primary Option Menu

Session

The session view (shown in Figure 18 on page 47) displays the following information in the status area:

- User ID
- Time
- Terminal
- Screen
- Language
- Application ID
- TSO logon
- TSO prefix
- System ID
- MVS account
- Release.

Note: **System ID** is a point-and-shoot field. **MVS Acct** and **Release** are point-and-shoot fields if over 8 characters long. Select these fields to display pop-up windows that contain additional information about the MVS account number and the ISPF environment.

MVS Acct
The account number identifying this MVS user.

System ID
Shows the **SYSPLEX** and **SYSNODE**.

SYSPLEX
The MVS sysplex name as found in the COUPLExx or LOADxx member of SYS1.PARMLIB.

SYSNODE
The network node name of your installation's JES.

Release
Displays the following variables:

- **ZOS390RL**— The OS/390 Release running on your system.
- **ZISPFOS**— The level of ISPF code that is running as part of OS/390 on your system. This might or might not match ZOS390RL.
- **ZENVIR**— The ISPF Environment description used by IBM personnel for internal purposes. The x.y numbers DO NOT directly correlate to an ISPF release number in all cases. Refer to the *OS/390 ISPF Dialog Developer's Guide* for a complete explanation.

```
 Menu  Utitlities Compilers  Options  Status  Help
-------------------------------------------------------------------------------
                          ISPF Primary Option Menu

0  Settings     Terminal and user parameters          User ID . : KEENE
1  View         Display source data or listings       Time.   . : 15:04
2  Edit         Create or change source data          Terminal. : 3278
3  Utilities    Perform utility functions             Screen. . : 1
4  Foreground   Interactive language processing       Language. : ENGLISH
5  Batch        Submit job for language processing    Appl ID . : ISR
6  Command      Enter TSO or Workstation commands     TSO logon : SERPROC
7  Dialog Test  Perform dialog testing                TSO prefix: KEENE
9  IBM Products IBM program development products      System ID : VS1C
10 SCLM         SW Configuration Library Manager      MVS acct. : 76TDOBOZ
11 Workplace    ISPF Object/Action Workplace          Release . : ISPF 5.2


     Enter X to Terminate using log/list defaults


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

Figure 18. ISPF Primary Option Menu Status Area – Session View

Function Keys

The function keys view (shown in Figure 19 on page 48) displays the following information in the status area:

Note: See the **Working with Function Keys and Keylists (The Function Keys Action Bar Choice)** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for information on changing these settings.

- Number of keys
- Keys displayed per line
- Primary range (lower or upper)
- Display set (primary or alternate)
- List name (name of the currently active keylist)
- List applid (application ID for the currently active keylist)
- List type (private or shared)
- Keylists (on or off).

```
 Menu  Utitlities Compilers  Options  Status  Help
──────────────────────────────────────────────────────────────────────────────
                          ISPF Primary Option Menu

0  Settings      Terminal and user parameters          No. of Keys: 24
1  View          Display source data or listings       Keys / Line: SIX
2  Edit          Create or change source data          Primary Set: SIX
3  Utilities     Perform utility functions             Display Set: SIX
4  Foreground    Interactive language processing       List Name: : ISRSAB
5  Batch         Submit job for language processing    List applid: ISR
6  Command       Enter TSO or Workstation commands     List type  : SHARED
7  Dialog Test   Perform dialog testing                Keylists   : PRIVATE
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager
11 Workplace     ISPF Object/Action Workplace
P  PDF Tools     PDF development tools
0  SER Tools     Local SER tools

     Enter X to Terminate using log/list defaults


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

Figure 19. ISPF Primary Option Menu Status Area – Function Keys View

Calendar

The calendar view (shown in Figure 20) displays the calendar for the current month in the status area.

```
 Menu  Utitlities Compilers  Options  Status  Help
──────────────────────────────────────────────────────────────────────────────
                          ISPF Primary Option Menu

0  Settings      Terminal and user parameters          <     Calendar     >
1  View          Display source data or listings          March     2001
2  Edit          Create or change source data          Su Mo Tu We Th Fr Sa
3  Utilities     Perform utility functions                            1  2  3
4  Foreground    Interactive language processing        4  5  6  7  8  9 10
5  Batch         Submit job for language processing     11 12 13 14 15 16 17
6  Command       Enter TSO or Workstation commands      18 19 20 21 22 23 24
7  Dialog Test   Perform dialog testing                 25 26 27 28 29 30 31
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager      Time . . . . : 02:55
11 Workplace     ISPF Object/Action Workplace          Day of year. :   086
P  PDF Tools     PDF development tools
0  SER Tools     Local SER tools

     Enter X to Terminate using log/list defaults


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

Figure 20. ISPF Primary Option Menu Status Area – Calendar View

All of the fields on the calendar are point-and-shoot fields that function as follows:

If you select	ISPF displays
<, calendar, or >	The previous, current, or next month, respectively.
Month	Calendar Month pop-up. Allows you to specify the month to be displayed.

Year Calendar Year pop-up. Allows you to specify the year to be displayed.

Day Calendar Start Day pop-up. Allows you to specify Saturday, Sunday, or Monday as the start day for the calendar.

Note: You can also display the Calendar Start Day pop-up by selecting Calendar start day... from the Options action bar choice on the ISPF Status pop-up window displayed when you select Status Area... from the Menu pull-down.

Date Julian Date pop-up. Provides the Julian date for the date selected.

Time Calendar Time Format pop-up. Allows you to specify a 12-hour or 24-hour time format for the calendar.

Day of year Standard Date pop-up. Provides the standard date for the date specified.

You can change the colors on the calendar by selecting **Calendar colors...** from the Options action bar choice on the ISPF Status pop-up window displayed when you select **Status Area...** from the Menu pull-down, as shown in Figure 21.

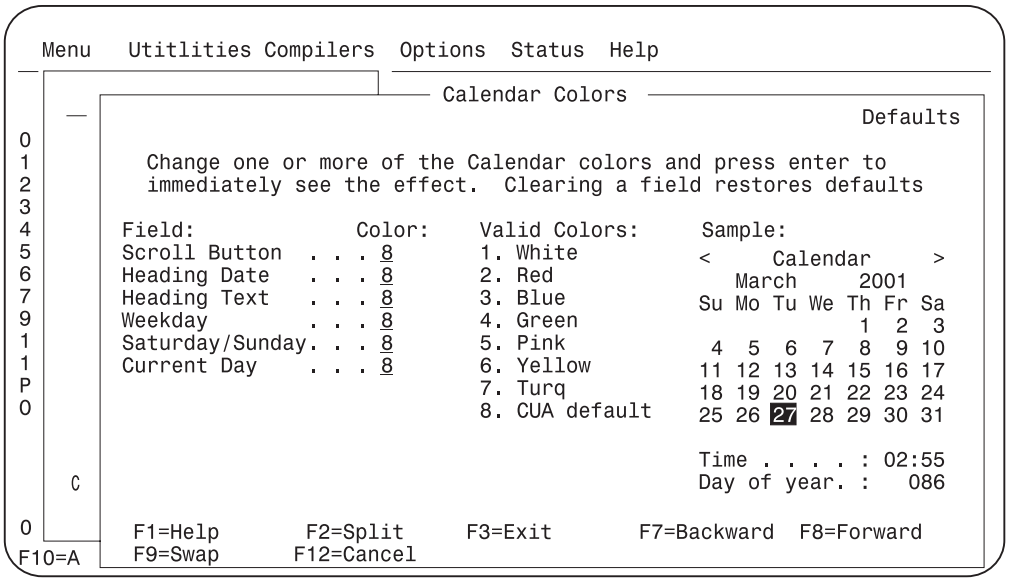


Figure 21. Calendar Colors Panel (ISPCALGC)

User Status

The user status view (shown in Figure 22 on page 50) displays the status you have defined from the Menu action bar pull-down.

Status Area on the Primary Option Menu

```
  Menu  Utitlities Compilers  Options  Status  Help
 ------------------------------------------------------------------------------
                             ISPF Primary Option Menu

 0  Settings      Terminal and user parameters            User ID       KEENE
 1  View          Display source data or listings         Time          14:57
 2  Edit          Create or change source data            Day           Tuesday
 3  Utilities     Perform utility functions
 4  Foreground    Interactive language processing
 5  Batch         Submit job for language processing
 6  Command       Enter TSO or Workstation commands
 7  Dialog Test   Perform dialog testing
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
 P  PDF Tools     PDF development tools
 O  SER Tools     Local SER tools

      Enter X to Terminate using log/list defaults


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

Figure 22. ISPF Primary Option Menu Status Area – User Status View

This status area was defined by selecting User status customization... (3) from the Options action bar choice on the ISPF Status pop-up window displayed when you select Status Area... (8) from the Menu pull-down, as shown in Figure 23.

```
   Menu  Utitlities Compilers  Options  Status  Help
 -  --   ----------------- User View Customization ----------------   ----------------
   |    |                                                          |
   |  _ |                                                          |
 0 |    |  Enter a desription and variable name below;             |  ID       KEENE
 1 |    |                                                          |           14:57
 2 |    |     Description:          Variable name:                 |           Tuesday
 3 |    |     User ID______         ZUSER___                       |
 4 |    |     Time_________         ZTIME___                       |
 5 |    |     Day__________         ZDAYOFWK                       |
 6 |    |     _____________         ________                       |
 7 |    |     _____________         ________                       |
 9 |    |     _____________         ________                       |
 1 |    |     _____________         ________                       |
 1 |    |     _____________         ________                       |
 P |    |     _____________         ________                       |
 O |    |     _____________         ________                       |
   |    |     _____________         ________                       |
   |    |                                                          |
   |  C |  Press EXIT or END to exit and save the changes.         |
   |    |  Press CANCEL to exit without saving the changes.        |
 0 |    |   F1=Help         F2=Split        F3=Exit                |
   --   |   F7=Backward     F8=Forward      F9=Swap                |  ard   F9=Swap
 F10=A  ------------------------------------------------------------
```

Figure 23. User View Customization Panel (ISPSAMUS)

User Point and Shoot

The user point-and-shoot view (shown in Figure 24 on page 51) displays the point-and-shoot function you have defined from the Menu action bar pull-down.

Note: If you are running in GUI mode, this function will be displayed as a push button.

```
 Menu  Utitlities Compilers  Options  Status  Help
──────────────────────────────────────────────────────────────────────────────
                          ISPF Primary Option Menu

0  Settings      Terminal and user parameters            Dialog test
1  View          Display source data or listings         Edit
2  Edit          Create or change source data            Edit Recovery Table
3  Utilities     Perform utility functions
4  Foreground    Interactive language processing
5  Batch         Submit job for language processing
6  Command       Enter TSO or Workstation commands
7  Dialog Test   Perform dialog testing
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager
11 Workplace     ISPF Object/Action Workplace
P  PDF Tools     PDF development tools
0  SER Tools     Local SER tools

     Enter X to Terminate using log/list defaults


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

Figure 24. ISPF Primary Option Menu Status Area – User Point-And-Shoot View

This status area, which contains a point-and-shoot field that invokes the Dialog Test function, was defined by selecting User point-and-shoot customization... (4) from the Options action bar choice on the ISPF Status pop-up window displayed when you select Status Area... (8) from the Menu pull-down, as shown in Figure 25.

```
  Menu  Utitlities Compilers  Options  Status  Help
─ ┌──────────────────────────── User Point-and-Shoot ──────────────────────┐
  │                                                                       │
0 │  Press EXIT or END to exit and save the changes.                      │
1 │  Press CANCEL to END to exit and save the changes.                    │
2 │                                                                       │
3 │  Enter Point-and-shoot text and SELECT keywords.                      │
4 │                                                                       │
5 │  Point-and-shoot text:  SELECT service parameters:                    │
6 │                                                            More:    + │
7 │  Dialog Test          PANEL(ISR@PRIM) OPTION(7) SUSPEND               │
8 │                       _                                               │
1 │                                                                       │
1 │                                                                       │
P │                                                                       │
0 │  Edit                 PGM(ISREDIT) PARM(P,ISREDMO1) NEWAPPL(ISR)      │
  │                                                                       │
  │                                                                       │
  │                                                                       │
  │                                                                       │
0 │   F1=Help       F2=Split      F3=Exit       F7=Backward    F8=Forward │
  └   F9=Swap       F12=Cancel                                            │
F10 └─────────────────────────────────────────────────────────────────────┘
```

Figure 25. User Point-And-Shoot Customization Panel (ISPSAMUP)

You can define up to nine point-and-shoot fields, which you can set to any SELECT service parameter. Refer to *ISPF Services Guide* for information on these parameters.

None

If you select None from the Status pull-down, nothing will be displayed in the status area.

Status Area Choice

When you select Status Area... from the Menu pull-down, ISPF displays the ISPF Status pop-up window (shown in Figure 26). The ISPF Status pop-up displays the current setting for the status area view. You can change this setting using either the Status choice on the action bar of this pop-up or the Status pull-down menu on the ISPF Primary Option Menu action bar.

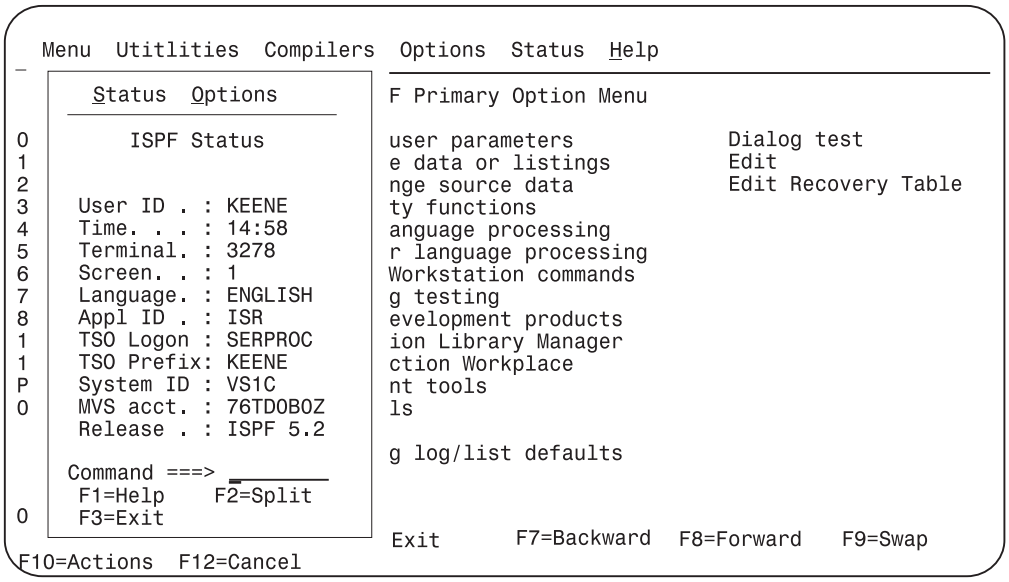


Figure 26. ISPF Status Pop-Up (ISPSAMMN)

The ISPF Status action bar choices function as follows:

Status The Status pull-down offers you the following choices:

> **Note:** See "Status Pull-Down" on page 45 for information on how to use these choices to change the view that is displayed in the status area.
>
> **1** **Session**
> **2** **Function keys**
> **3** **Calendar**
> **4** **User status**
> **5** **User point and shoot**
> **6** **None**
>
> **Note:** The current setting is shown as an unavailable choice; that is, it displays in blue (the default) with an asterisk as the first digit of the selection number (if you are running in GUI mode, the choice is *grayed*).

Options

The Options pull-down offers you the following choices:

> **1** **Calendar start day...** Displays the Calendar Start Day pop-up, where you can specify Saturday, Sunday, or Monday as the start day for the calendar.
>
> **Note:** You can also display this pop-up by selecting a day of the week if you have the calendar displayed in the status area.

2 **Calendar colors...** Displays the Calendar Colors pop-up, where you can change the colors on the calendar. See Figure 21 on page 49 for an example.

3 **User status customization...** Displays the User View Customization pop-up, where you can define what you want displayed in the status area. See Figure 23 on page 50 for an example.

4 **User point and shoot customization...** Displays the User Point and Shoot Customization pop-up, where you can define point-and-shoot fields to be displayed in the status area. See Figure 25 on page 51 for an example.

Understanding ISPF Panels

A *panel* is a predefined display image that you see on a display screen. ISPF formats all panels to fit on a 24-line by 80-character screen. On a 3278 Model 3 or 4, data that you can scroll occupies the full length of the screen (32 or 43 lines). On a 3278 Model 5, ISPF normally displays information in *default mode*; that is, 24 lines by 80 characters, with the same size characters as other models. Browse and edit data that is wider than 80 characters is displayed with the smaller *native mode* characters, that is, up to 132 per line. You can use the Settings option (0) to override the automatic switching of modes.

Panel Format

Figure 27 shows how ISPF formats the first three and last two lines of each display:

```
Action Bar
--------------------------------------------------------------------------
                               Title                              Short Message


Long Message
Option  ===>                                                  Scroll ===>
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
```

Figure 27. Panel Format

Note: The **Panel display CUA mode** field on the ISPF Settings panel determines where the Command or Option line and long messages are displayed. The default setting selects **Panel display CUA mode**, which causes the Command or Option line to be displayed on the bottom of the panel. The default setting also selects **Long message in pop-up**, which causes long messages to be displayed in a pop-up window directly above the Command or Option line. To display the command or option line and long messages at the top of the panel, select option 0, deselect the **Panel display CUA mode** field, and deselect the **Command line at bottom** field. Refer to *ISPF Dialog Developer's Guide and Reference* if you need more information about the **Panel display CUA mode** and **Command line placement** fields.

The fields on Figure 27 function as follows:

Action Bar
The action bar provides access to pull-down menus that give you a new

and faster way to move around in the product as well as access to command stacking. See Figure 1 on page 5 for more information about using the action bar.

Title Identifies the function being carried out and, where appropriate, the library or data set information.

Short Message
Shows:

- Current line for Browse, and column positions for View, Browse, and Edit
- Current row position in a table display if the short message area is not overlaid by a function-requested message
- Successful completion of a processing function
- Error conditions (with an audible alarm, if one is installed). Refer to *ISPF Dialog Developer's Guide and Reference* for information about coding an alarm.

Command/Option
Allows you to enter a command or, on a menu, to enter either a command or an option.

Scroll Indicates the scroll amount (if scrolling applies). You can type over it to change it. The valid scroll amounts are 0–9999, PAGE, HALF, MAX, CSR, and DATA.

Long Message
Displays an explanation of error conditions in a pop-up window when you enter the HELP command (see "Getting Help" on page 58). On some displays, data may be overlaid temporarily by a long message.

Function Keys
Displays settings for the function keys. These settings are controlled through the Function keys pull-down on the action bar on the ISPF Settings panel.

Panel Types

When using ISPF, you see three basic types of panels:

- Menus (selection panels)
- Data-entry panels
- Scrollable data displays.

Menu

A *menu*, or selection panel, allows you to type a number or letter in the **Option** field and press Enter to select one of the listed items. The number or letter can be typed in either uppercase or lowercase. Allowable numbers and letters are shown in high intensity. You can also enter ISPF commands. See Figure 16 on page 42 for an example of a menu.

Note: If the word `BLANK` or `blank` is listed, leave the **Option** field blank and press Enter to select that option.Do not type the word `blank`.

Data-Entry Panels

A data-entry panel is a panel on which you specify information, such as data set names, job statement parameters, and language processing options. If you do not enter a required value or if you enter inconsistent values, ISPF prompts you with a message.

Some data-entry fields retain their previous values. If so, the next time you use the panel, you do not have to type them again. Just press Enter. If you do not want those values, type over them and then press Enter.

The retained values come from your user profile, which ISPF automatically builds and maintains across sessions. See "User Profiles" on page 59 for more information about user profiles.

Edit modes and defaults are also maintained in the Edit portion of your user profile. Refer to *ISPF Edit and Edit Macros* for more information.

Entering Commands in ISPF

ISPF provides flexibility by accepting various types of commands and having many methods for entering them. Table 1 provides an overview of the entry methods and commands available.

Table 1. Entry Methods and Command Types

Entry Methods	TSO Cmds, CLISTs, and REXX EXECs	DM Cmds	PDF Primary Cmds	PDF Line Cmds	WS Cmds
ISPF Command Shell (option 6)	X				X
Option field	X	X			X (1)
Command field	X	X	X		X (1)
Line Command fields (1–6 characters)				X	
Line Command fields (9 characters)	X			X	
Note: (1) Available when prefixed by "WS"					

For example, to use the TSO ALLOCATE command, you could enter:

```
Command ===> TSO ALLOCATE
```

Also, CLIST names and REXX EXEC names can be preceded by a percent (%) symbol, as in:

```
Option ===> TSO %CLIST
```

This symbol informs TSO that the command is a CLIST or REXX EXEC, not a TSO command.

You can also use two command entry methods not shown in the table: the PA keys and the function keys. PA1 (ATTENTION) and PA2 (RESHOW) are hardware keys that you cannot redefine. You can use function keys to enter all commands.

Types of Commands

Three types of commands operate at three levels:

- TSO commands, CLISTs, and REXX EXECs

You invoke high-level commands through the MVS/TSO operating system. These include TSO commands, CLISTs, REXX EXECs, and the commands assigned to the PA1 and PA2 keys.

- DM component commands

  You call middle-level commands through the DM component of ISPF.

- PDF component primary and line commands

  You call low-level commands through the PDF component of ISPF.

TSO Commands, CLISTs, and REXX EXECs

ISPF gives you access to the MVS/TSO operating system by letting you enter TSO commands, CLISTs, and REXX EXECs from within ISPF. *TSO Extensions Command Language Reference* and *MVS/XA TSO Extensions Command Language Reference* contain descriptions of all TSO commands.

You can enter most TSO commands, CLISTs, and REXX EXECs from all three panel types—data entry, menu, and scrollable data display. However, some TSO commands, such as LOGON and LOGOFF, can cause unwanted results when you enter them from ISPF. This is also true of CLISTs and REXX EXECs that contain these TSO commands. See the **Rules for Entering TSO Commands** and the **Rules for Entering CLISTs and REXX EXECS** sections of the ***Command (Option 6)*** chapter of the *ISPF User's Guide Volume II* for more information.

DM Component Commands

DM component commands are valid from all three types of panels. However, the validity of some commands depends on the type of panel displayed or the type of terminal you use. For example, the scroll commands—UP, DOWN, LEFT, and RIGHT—are valid only on scrollable data displays, and the SPLITV command is valid only on a 3290 display terminal.

For a description of the ISPF commands, default function key settings, and the PA1 and PA2 keys, see "ISPF System Commands" on page 88.

PDF Component Primary and Line Commands

These commands are valid only when you call them within the PDF component. Exceptions are the primary and line commands for Edit (option 2), which are defined in *ISPF Edit and Edit Macros* those for LM Facility (option 8), which are defined in *ISPF Library Management Facility* and those for SCLM (option 10), which are defined in *ISPF Software Configuration and Library Manager (SCLM) Developer's and Project Manager's Guide* The ISPF primary option that you are using determines which commands are valid. For example, the CAPS primary command is valid only when using the Edit option.

See the chapter in this book on the option you are using for information about the valid commands for that option.

PDF component primary commands can affect a complete data set or PDS member, but the line commands can affect only one or more specified lines within a data set (unless you specify all the lines in the data set). For example, the Edit COPY command can copy a complete data set, whereas the C or CC Edit line command can copy only lines or blocks of lines within a data set.

ISPF Command Syntax Notation

The notation conventions for ISPF command syntax follow.

- Uppercase commands and their uppercase parameters show required entry.
- Lowercase parameters show variables (substitute your values for them).

- Brackets([]) show optional parameters (required parameters do not have brackets).
- An OR (|) symbol shows two or more parameters you *must* select from.
- Stacked parameters show two or more parameters you *can* select from.
- Braces ({}) with stacked parameters show that you *must* select one.
- Underscores show defaults.

Stacking Commands

To enter more than one command, you can stack them by typing a special delimiter between them. The default delimiter is a semicolon. Use the Settings option (0) to change the delimiter.For example, to stack two Edit CHANGE commands, use:

```
Command ===> CHANGE ALL ABC XYZ;CHANGE ALL PQR GHIJK
```

The system variable for the delimiter is ZDEL. Refer to the *ISPF Dialog Developer's Guide*, Appendix E, for more information about ZDEL.

Dual Command Processing

You can enter information on a command or option line in combination with pressing a related function key. The command is called first. For example, typing `4` on the command line and pressing F7 (UP command) is the same as typing `UP 4` on the command line and pressing Enter.

If the command you type is unrelated to the command assigned to a function key you press, ISPF passes the entry to the function in control, which either processes or ignores the entry. For example, if the Edit function is in control, ISPF may display an error message. However, if the Tutorial is in control, the command is ignored. ISPF processes any stacked valid commands.

Line Command Fields

Line command fields can take many forms. Some have headings, some do not. Most are blank, but some contain single quotation marks or sequence numbers. For some, you type one character; for others, you type up to nine characters (even typing over data set names). Table 2 shows the functions that provide the line command fields.

Table 2. Characteristics of the Line Command Field

Functions Providing Line Command Fields	Heading Displayed	Characters Allowed	Initial Contents
Member Selection List	None	1–9	Blank
Edit (option 2)	None	6	Quotes or numbers
Data Set List utility (option 3.4)	Command	9	Blank
Command Table utility (option 3.9)	None	4	Quotes
Format Specification utility (option 3.11)	None	1	Blank
SCRIPT/VS (option 4.9)	Line Cmd	1	Quotes
Dialog Test (option 7): Variables (option 7.3) Tables (option 7.4) Traces (option 7.7) Breakpoints (option 7.8)	None	4	Quotes

The line command field for member selection lists is blank and has no heading. This includes typical member lists, which have a one-character line command field, and the member list displayed when you use option M of the Data Set List utility, which has a nine-character line command field.

Quotes appear when you create a new data set or member, or when you insert one or more lines. Sequence numbers appear if you have NUMBER ON in your Edit profile.

Data set lists with nine-character line command fields allow you to type over data set names, thus extending the length of the fields to allow you to type long TSO commands, CLIST names, and REXX EXEC names. See the **TSO Commands, CLISTs, and REXX EXECs** section of the ***Data Set List Utility (Option 3.4)*** in the *ISPF User's Guide Volume II* for information about entering TSO commands, CLISTs, and REXX EXECs in a line command field. Also, the **Rules for Entering TSO Commands** and the **Rules for Entering CLISTs and REXX EXECS** sections of the ***Command (Option 6)*** chapter of the *ISPF User's Guide Volume II* contain information about the rules governing the entry of TSO commands, CLISTs, and REXX EXECs from within ISPF.

PDF Component Line Commands

Most PDF component line commands use only one letter, such as S, for selecting a member from a member list. Others, such as many of the Edit line commands, use more than one letter and sometimes allow you to add a number so the command affects more than one line. For example, the UC line command plus the number 3, as in `UC3`, converts three lines to uppercase.

Another type of line command is the *block* line command, which affects the block of lines between and including the lines on which the commands are entered. For example, the UCC line command, when entered beside two different lines, converts all lines between and including the two commands to uppercase.

Getting Help

The HELP command (F1/13) shows you general information about an ISPF system command, ISPF option, or panel, or offers additional information about a message that has been displayed in the upper-right corner of the screen.

For short messages, HELP displays a one-line explanation. To get further information, enter the HELP command a second time for the appropriate section of the tutorial. Long messages display (by default) in a pop-up window. Enter `END` (F3/15) or `RETURN` (F4/16) to return to the screen that you were viewing when you entered the HELP command.

For *some* messages in LM Facility (option 8 on the ISPF Primary Option menu), entering HELP a second time displays a message help panel, not the tutorial. It describes what error condition may have caused the message and how to correct the error.

Ending an ISPF Function or ISPF

To end an ISPF function without ending ISPF, enter `END` or `RETURN`.

Notes:

1. If you are using the Hardcopy utility (option 3.6) or the Batch option (option 5), `END` or `RETURN` submits your job for processing. Type `CANCEL` (or press F12) to leave the Hardcopy utility without submitting a job.
2. If a SuperC or Search-For member list is displayed, `END` processes any members you have selected. Enter `CANCEL` or `RETURN` to leave one of these member lists without processing your selections.

To end ISPF from the ISPF Primary Option Menu, you can use the:
- EXIT command (F3)
- END command
- RETURN command
- Exit option (X).

If the display screen is split, taking one of the actions listed above ends ISPF on the active logical screen only. Refer to *ISPF Dialog Developer's Guide and Reference* for more information.

Exit Option (X)

The Exit option ends ISPF using any defaults for processing log and list data sets that you have specified using the Log/List pull-down on the ISPF Settings panel action bar. If you have not specified any defaults and if a log or list data set has been generated, the Exit option displays the Specify Disposition of Log and List Data Sets panel.

Jump Function

If you are not in split-screen mode, you can use the *jump function* to immediately leave ISPF from any panel by entering `=X`. There is one exception.If you are using any of the Dialog Test options (7.1–7.T) or the SCLM options (10.1–10.6), entering `=X` returns you to the ISPF Primary Option Menu. If you are at the Dialog Test or SCLMPrimary Option Menu, enter either `X` or `=X` to return to the ISPF Primary Option Menu. Enter `X` or `=X` to end ISPF.

User Profiles

ISPF stores information in your user profile. This allows ISPF to insert a value in panel input fields by using the values you last entered on either the same panel or a similar type of panel. Sometimes default values are provided if you have not specified otherwise. Information maintained in your user profile includes:

- Project name, group names, and type
- Job statement information [1]
- SYSOUT class for printed output [1]
- Defaults for list and log allocation and processing
- Terminal characteristics and function key definitions
- Edit profiles, including mask, tabs, and bounds
- Current scroll amount for Browse, Edit, Data Set List, and member lists [1]
- Processing options for each of the language processors
- Data set allocation/information parameters.

1. This information is maintained separately for SCLM.

ISPF maintains this information automatically from one session to another. If you are a new user, you have to enter certain information the first time. But then, you simply review the information and make whatever changes you want before proceeding.

ISPF maintains sets of job statements for:

- JCL for printing the Log and List data sets
- The Hardcopy utility (3.6)
- The Outlist utility (3.8)
- The SuperC utility (3.12)
- The SuperCE utility (3.13)
- The Foreground Print Options panel (option 4, excluding SCRIPT/VS)
- The Batch option (5)
- The SCLM option (10)
- SCRIPT/VS draft output (4.9)
- SCRIPT/VS final output (4.9)
- Ending ISPF.

Thus, you can provide different job statement parameters for each of these functions. For more information, see "Job Statement Information" on page 160.

Splitting the Screen Horizontally or Vertically

At any time during a dialog, you can partition the physical display screen into two or more *logical* screens (ISPF enables a maximum of 32) by using the SPLIT command. *The logical screens are treated as though they are independent ISPF sessions.* The maximum number of screens available to you is dependant on your installation's MAXIMUM_NUMBER_OF_SPLIT_SCREENS keyword value entry in the ISPF Configuration table. ISPF ships with a default figure of 8. Support for 32 split screens is available for all terminal types except the 3290. Support for 3290 terminals has not changed.

Note: If you are running in GUI mode, when you split a screen, the new screen is displayed as a new physical window. If you want, representations of every screen can be displayed at once. Although a 3270 screen can only display two screens at one time, there can be other screens (up to 32) that are not visible. You can select which logical screen to display by using the SWAP LIST command to display a list of logical screens.

The SPLIT Command

You enter split-screen mode by using the SPLIT command. You also use this command to reposition the horizontal line that separates the two logical screens on a 3270 display. On a 3270 display the location of the cursor identifies the active logical screen. On a 3270 display, the horizontal divider line that separates the logical screens is not considered part of either logical screen. If the cursor is placed on this horizontal divider line and a function key is pressed, the result is the same as if the ENTER key was pressed and the cursor is positioned on the active logical screen's command line. Because ISPF runs in both host and GUI modes, the SPLIT command behaves differently in each.

SPLIT command without parameters, in 3270 mode:

If only one screen is currently being used, the physical display is divided into two logical screens with a divider at the cursor. If two or more screens exist, the divider line is moved, but no new screen is started.

SPLIT command without parameters, in GUI mode:

A new logical screen is added each time the command is given, until the maximum number is reached. After the limit is reached, a message appears when the command is issued again.

SPLIT NEW command, in 3270 mode:

A new logical screen is added each time the command is given, until the maximum number is reached. After the limit is reached, a message appears when the command is issued again. Each new logical screen is added below the cursor, where the split line appears. If two or more screens already exist, the new one replaces the screen in which the SPLIT command was *not* entered.

SPLIT NEW command, in GUI mode:

This command behaves the same as the SPLIT command without parameters.

End split-screen mode by ending the application on all but one logical screen. The remaining logical screen is then expanded to the full size of the display screen.

The SWAP Command

Although you can alternately use any logical screen, only one of the logical screens is considered active at a time. The location of the cursor identifies the active screen. You make a screen active by using the SWAP command and its parameters to choose the desired screen.

The parameters on the SWAP command–LIST, PREV, NEXT, screen_name, and n–control which screens you see displayed.

SWAP command without parameters in 3270 mode:

If only one screen exists, this command has no effect. If more than one screen exists, this command moves the cursor between the two logical screens that are displayed.

SWAP command without parameters in GUI mode:

If only two screens are present, the cursor moves to the one that it is not currently on. If more than two screens exist, this command defaults to the SWAP LIST command and causes the ISPF task list to display in a pop-up window. See SWAP LIST command in GUI mode on page 62 for more information.

SWAP PREV | NEXT | screen_name | n commands in both modes:

Entering SWAP PREV changes the focus (for GUI mode) or display (for 3270 mode) to the next lower screen number from the one where the command is entered. Repeatedly issuing the same command causes each lower-numbered screen to display until screen number 1 is reached, then the counter wraps back to screen number 32 (or your installation's maximum number).

Entering SWAP NEXT changes the focus (for GUI mode) or display (for 3270 mode) to the next higher screen number from the one where the command is entered. Repeatedly issuing the same command causes each higher-numbered screen to display until screen number 32 (or your maximum) is reached, then the counter wraps back to screen number 1.

Entering SWAP screen_name changes the focus (for GUI mode) or display (for 3270 mode) to the screen named screen_name if it is active.

Entering SWAP *n* changes the focus (for GUI mode) or display (for 3270 mode) to the screen numbered *n* (ZSCREEN variable) if it is active.

SWAP LIST command in 3270 mode:

This command displays the ISPF Task List, from which you can select which screen to display. The screen you select replaces the screen on which you entered the command.

SWAP LIST command in GUI mode:

This command displays the ISPF Task List (Figure 28) , from which you can select which screen to focus on.

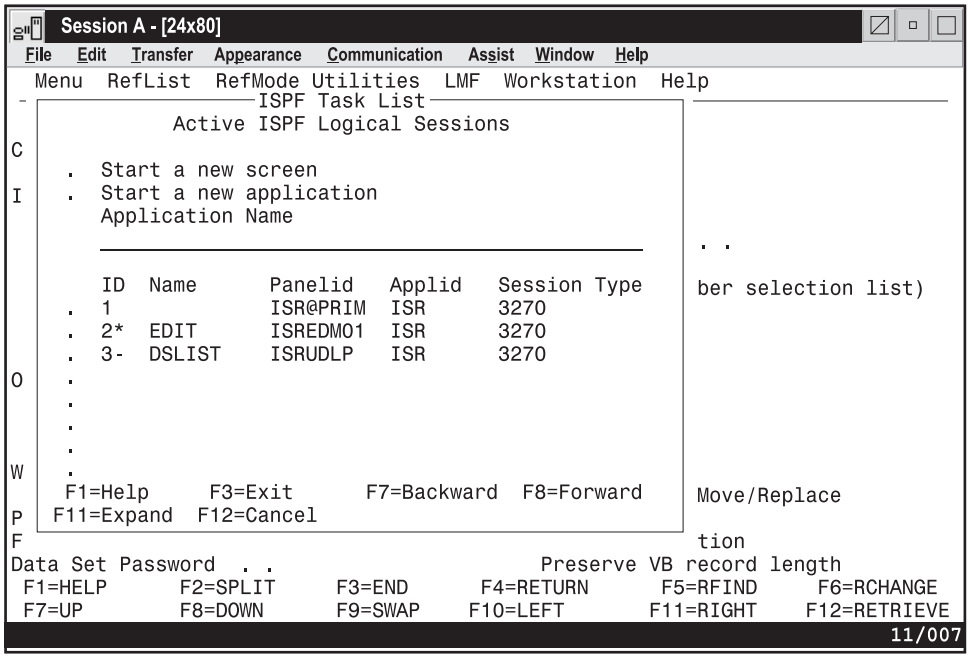


Figure 28. ISPF Task List

All of the fields on the ISPF Task List panel are point-and-shoot fields. The results when choosing one are as follows:

Start a new screen
Starts a new logical ISPF screen.

Start a new application
This field is used in conjunction with the **Application Name** field. If you choose ″Start a new application″ you ***must*** enter an application name in the ″Application Name″ field.

Application Name

The name of an application you want to start by choosing the ″Start new application″ field on the ISPF Task List panel. This application is started in a new logical screen. ISPF invokes the application through the **ISPF START** command, so any application name and parameters that are valid for the START command are valid in the Applcation Name input field. If you need more space to enter the application name and parameters, press the Expand PF key to display a pop-up window that contains a longer input field.

For example, if a user types ″keylist″ in the Application Name input field and presses Enter, the ISPF KEYLIST application is invoked in a new logical screen.

Select a screen from the list of active sessions

Provides a list of active sessions for you to choose from.

The SWITCH Command

In addition to using the swap command to change the focus of your session from one screen to another, if you are running the ISPF Client/Server you can use the SWITCH command to change the way the screen is displayed, either GUI mode or 3270 mode.

From any GUI screen you can enter command **SWITCH 3270**. The GUI screen disappears and the screen is then displayed fullscreen on your 3270 display. The 3270 screen is now operational. If you had been in split screen mode, the remaining GUI screens remain in GUI, but are disabled. To swap back and forth between the GUI and 3270 sessions now, you must use the SWAP commands (SWAP x, SWAP list, and so forth). The SWAP List panels ISPSLIST and ISPTLIST have a field indicating the session type (GUI or 3270).

If you want to switch your 3270 screen back to GUI mode, enter **SWITCH GUI** on the command line.

Rules for Splitting and Swapping 3270 and GUI Screens

The rules for splitting and swapping screens when you have combinations of both 3270 and GUI screens are as follows:

- If you have only one 3270 screen and enter ***split***, the new logical screen is displayed in GUI mode.
- If you have two or more 3270 screens and enter ***split*** from one of them, the next 3270 screen is displayed fullscreen, but no new screen is started.
- If you enter ***split*** from a GUI screen, a new GUI screen is started (until the maximum number is reached).
- If you enter ***split new*** from either a GUI or 3270 screen, the new logical screen is displayed in GUI mode.
- If you have one 3270 screen and one GUI screen, then entering ***swap*** from either of these screens causes a swap to the other.
- If you have two GUI screens and one 3270 screen, ***swap*** from one of the GUI screens swaps you to the other GUI screen. ***Swap*** from the 3270 screen does nothing.
- If you have one GUI screen and two 3270 screens, ***swap*** from one of the 3270 screens swaps to the other 3270 screen. ***Swap*** from the GUI screen does nothing.

- If you have two GUI screens and two 3270 screens, ***swap*** from one of the GUI screens swaps you to the other GUI screen. ***Swap*** from one of the 3270 screens swaps you to the other 3270 screen.
- If you have more than two GUI screens and enter ***swap*** from one of them, the **SWAP LIST** panel appears.
- If you have more than two 3270 screens and ***swap*** from one of them, you swap to the next 3270 screen.
- Entering **SWAP PREV | NEXT | screen_name | n** commands in either GUI or 3270 mode swaps as described in "The SWAP Command" on page 61.
- After a GUI screen is enabled, you can swap to other GUI screens using your mouse pointer.

SWITCH Command Restrictions

At the completion of ISPF command processing, ISPF does not reprocess the panel from which the command was invoked. The panel is simply relaunched to the screen without processing, for example, the panel's)INIT section. Therefore, if certain constructs are defined within the panel sections based on the ZGUI variable (non-blank indicating connected in GUI mode), these constructs are not defined properly until after the Enter key is pressed following the SWITCH invocation.

The following are some known restrictions for the SWITCH command:

- When switching to GUI mode, the first panel displayed in GUI mode cannot contain group boxes and images that are defined on the panel. After the user presses the Enter key, causing the panel to be reprocessed, then these constructs are visible in GUI mode.
- When switching to 3270 mode, the name of any group boxes defined on the panel that the SWITCH was issued from display on the screen in 3270 mode. After the user presses the Enter key, causing the panel to be reprocessed, these group box names disappear and any panel text under these names reappears.
- If a 3270 screen already exists when the SWITCH 3270 command is invoked from another GUI screen, the screen from which the SWITCH was invoked is displayed in the full 3270 emulator session without a split line, regardless of the user's ***Always show split line*** setting. The other 3270 sessions are hidden and available for display if the appropriate swap command is entered.
- If pop-up windows are displayed in GUI mode when the SWITCH 3270 command is invoked, those pop-up windows are suspended on the 3270 session and the panels are displayed as fullscreen panels. If new addpops are then invoked, these new panels are displayed as pop-ups.
- The user cannot switch to 3270 mode when running in ***batch*** GUI mode.
- When running in GUI mode, if at least one screen has been switched to 3270 mode, then the user is not allowed to change the screen format in ISPF Settings.
- If you started GUI mode from ISPSTART and specified **GUISCRD** or **GUISCRW** values different than your actual emulator session, then you are not allowed to switch to 3270 mode.

3290 Terminals

On 3290 terminals, in addition to splitting the screen horizontally, you can use the SPLITV command to split the screen vertically, for a total of four logical screens. In the case of the 3290 terminal, four is the maximum number of screens possible. (The SPLITV function is not active if the data being displayed on a screen is more than 80 characters wide.) Figure 29 shows the effect of SPLIT and SPLITV, starting in single-screen mode.

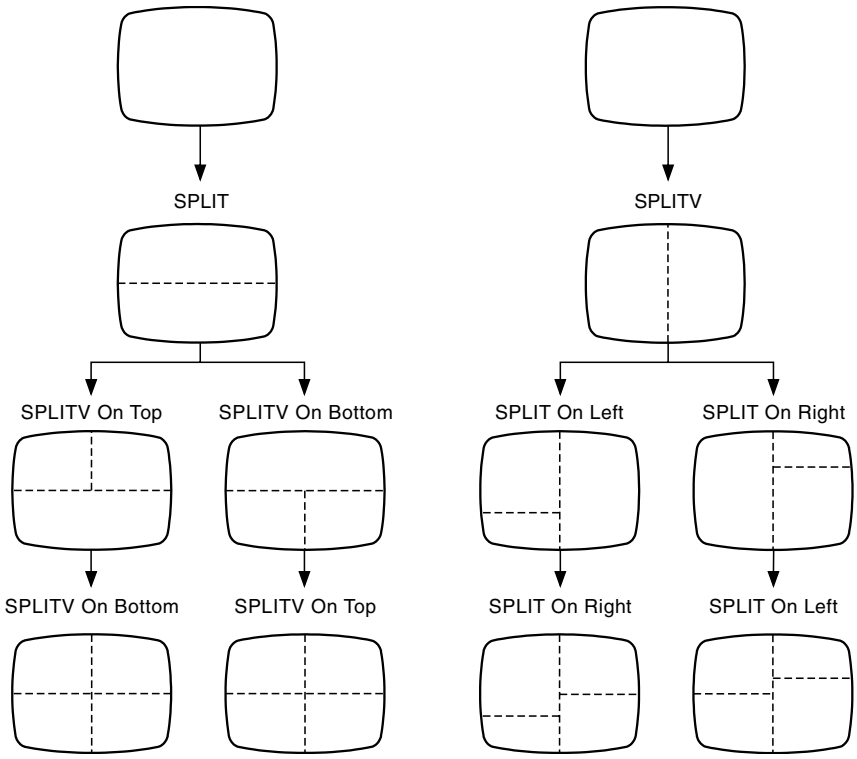


Figure 29. Splitting the 3290 Screen

Notes:

1. ISPF logical screens are separate subtasks that do not share subpool 0 (attached with SZERO=NO parameter.) Thus, VSAM data sets cannot be accessed from more than one logical screen.
2. If you are in a VSAM application, perform a split screen operation, then enter another VSAM application in the second session, you must be sure to end the second session before you end the first session, or an abend can occur.
3. On 3290 hardware, using the jump function to move from screen to screen might result in the loss of data that has been typed but not processed. The use of the 3290 hardware jump is not recommended.
4. In split-screen mode, if you type a command on the command line and swap screens before pressing Enter, the command is erased.
5. You cannot start ISPF in GUI mode if you are configured to run ISPF in 3290 partition mode.

National Language Support

National Language Support (NLS) gives countries the option of translating commands and keywords so that users can enter them in the country's national language. In addition, panels, messages, and literal modules are shipped with the product. Languages supported are German, Swiss-German, and Japanese.

When the Danish, French, Korean, Traditional Chinese, Simplified Chinese, Spanish, Brazilian-Portuguese, or Italian session language is specified, its respective literal module is used. However, the ISPF product panels and messages are displayed in English.

Chapter 3. Using Personal Data Set Lists and Library Lists

Personal lists are named lists of data sets and ISPF libraries that you can use to speed up access to frequently used data sets. You can use personal lists to fill in panel fields quickly and to create data set lists that are built from more than one level name. **Personal data set lists** contain data set names and volumes. **Personal library lists** contain lists of ISPF library names and concatenations.

ISPF provides four types of personal lists:

Personal data set list
: Lists of up to 30 data set names. Each name can include a member name or a volume name, or both. Personal data set lists can also contain workstation file names and data set name levels. See "Personal Data Set Lists" on page 69.

Personal library list
: Lists of up to eight ISPF library names or ISPF library concatenations. ISPF library names contain three qualifiers called ***project***, ***group***, and ***type***. Personal list entries can optionally contain a member name. See "Personal Library Lists" on page 70.

Reference data set list
: A special kind of personal data set list in which ISPF saves the names of the most recently used data sets. This list is always named REFLIST. See "Reference Data Set List" on page 70.

Reference library list
: A special kind of personal library in which ISPF saves the names of the most recently used ISPF libraries. This list is always named REFLIST. See "Reference Library List" on page 70.

You can have a personal data set list with the same name as a personal library list. ISPF reserves the name REFLIST as the name of the reference lists, so there is a personal data set list called REFLIST, and a personal library list called REFLIST.

Reference lists are active lists of data sets and libraries that you have referenced in your ISPF session. ISPF adds a data set name to the data set reference list when you enter a data set name in the **Other Partitioned or Sequential Data Set Name** field. A library is added to the library reference list when you enter a library in the **ISPF Library** field. Only data sets and libraries that are successfully allocated by ISPF's ALLOCATE routine are added to the reference lists.

Note: Reference lists can be manipulated just like any other personal list, but ISPF might dynamically change reference lists when new data sets or libraries are referenced by ISPF.

Current lists are the most recently opened or the last list to which something was saved from within the personal list panels. One named data set list, and one named library list, are the current lists at any time. The current list is used for the **NRETRIEV** key, and in the RefList pull-downs. The current list names are shown in the RefList pull-down choices, and in the lists of personal lists.

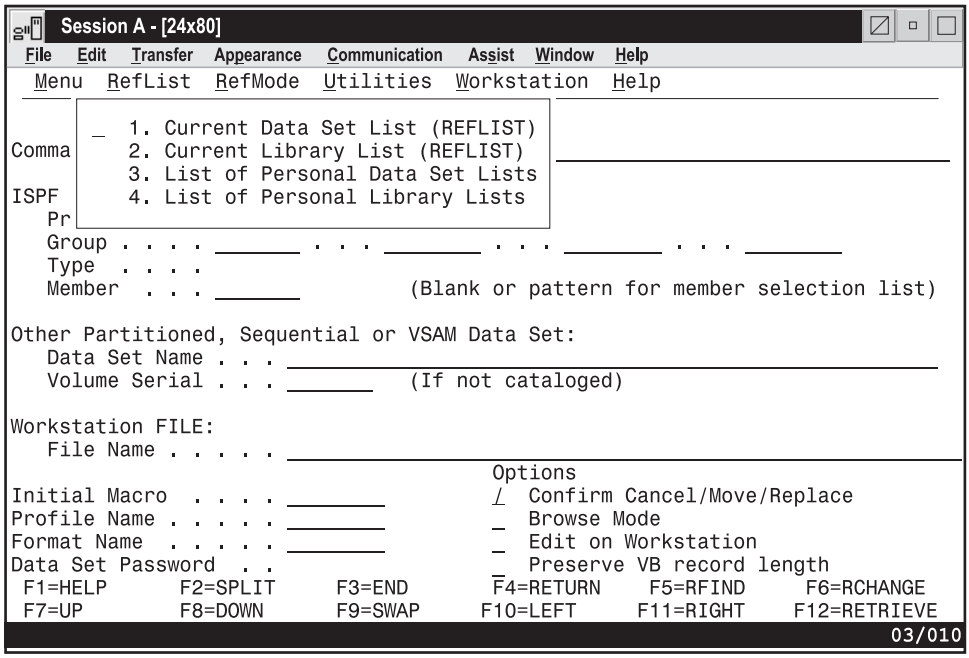


Figure 30. RefList Pull-Down Menu

Note: The RefList pull-down is unavailable from Utilities options 8, 9, and 13; it does not offer library list choices from options 4 and 6.

You can use personal lists for data set selection from the View Entry, Edit Entry, and most of the Utilities panels. You can access lists in two ways:

- From the RefList pull-down menu on the action bar of the View Entry, Edit Entry, and most Utilities panels, as shown in Figure 30.
- Through the command interface (or fast path) described in "Command Interface (Fast Path) to the Personal List Function" on page 80.

Personal List Modes

The action taken when you select a data set from a list depends on how you have the Mode set. All personal lists can be set to either Retrieve or Execute mode from the RefMode pull-down on the action bar of the View Entry, Edit Entry, and most Utilities panels, as shown in Figure 31 on page 69. List Retrieve displays in blue (the default) with an asterisk as the first digit of the selection number (if you are running in GUI mode, the choice is *grayed*), which indicates that RefMode is currently set to Retrieve.

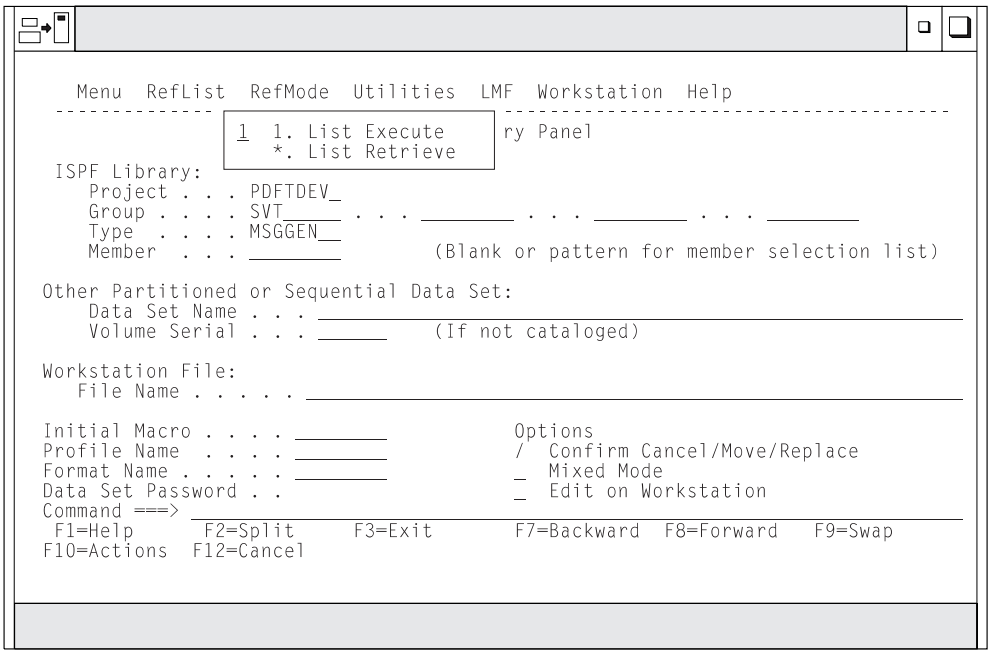


Figure 31. RefMode Pull-Down Menu

The RefMode pull-down offers you the following choices:

List Execute
: Sets personal data set lists and personal library lists to Execute mode; that is, when you select an entry from the list, the information is placed into the **ISPF Library** or **Other Data Set Name** field, and ISPF proceeds as if you also pressed the Enter key.

List Retrieve
: Sets personal data set lists and personal library lists to Retrieve mode; that is, when you select an entry from the list, the information is placed into the **ISPF Library** or **Other Data Set Name** field, but the simulated pressing of the Enter key is not performed. This allows you to set other options before you press Enter.

Note: These modes only effect selections from the personal list dialogs and the REFACTD and REFACTL fast path commands. They ***do not*** have any effect on the operation of the NRETRIEV command. NRETRIEV always acts like a List Retrieve.

Personal Data Set Lists

You can build lists of personal data sets that contain up to 30 data set names. You can have as many lists as you like as long as each has a unique name. Personal data set lists are a good way to group (by project, for example) those data sets that you use frequently. You can use personal data set lists to avoid typing in data set names and to create customized data set lists similar to those using ISPF Option 3.4. To create or use a personal data set list, do one of the following:

- Select the Personal Data Set List choice from the RefList pull-down on the View Entry, Edit Entry, or Utilities panel action bar
- Enter the REFACTD command from the command line.
- Use the "Save As" action when viewing the data set reference list (REFLISTD command).

Using Personal Data Set Lists

The REFACTD command displays the Personal Data Set List panel, as shown in Figure 32 on page 72. If you have one or more personal data set lists, ISPF displays the current list. If you have no personal data set lists, ISPF displays the reference list called REFLIST, which is updated by ISPF whenever a new data set is used by ISPF.

Personal Library Lists

You can build personal library lists of up to eight ISPF libraries. You can have as many lists as you like as long as each has a unique name. Personal library lists are a good way to group (by project, for example) those libraries that you use frequently. To create or use a personal library list, select the Personal Library List choice from the RefList pull-down on the View Entry, Edit Entry, or Utilities panel action bar. You can also use the REFOPENL command from any ISPF command line and use the Save As action to save the reference library list as a personal library list.

Note: Personal library lists are not available from RefList pull-downs for any options that do not support library names. For example, the data set list utility and data set print utilities do not support personal library lists.

Personal library list can be used with the NRETRIEV name retrieve command.

Personal library lists are used, created, and maintained with facilities similar to those of personal data set lists. Select the library list options from the RefList pull-downs to access the library list functions. As with personal data set lists, the list named REFLIST is the reference list that contains the most recently referenced ISPF libraries and library concatenations.

Reference Data Set List

The Reference Data Set List is a special personal data set list that contains a list of up to 30 data set names (and the volumes on which they are located) that you have referenced (that is, entered on panels or called with services) throughout ISPF. Data set names are also added to the list when ISPF refers to them; for example, during a MOVE/COPY operation or a DELETE function.

The reference data set list is a personal data set list with the name REFLIST. The name REFLIST is reserved by ISPF to refer to the reference list, but you can use the list just like any other list. If you save a personal list under the name REFLIST, the reference list reflects the names you save into it, but it is still updated when other data set are referenced by ISPF.

You can turn off the automatic updating of reference lists by using the settings panel on any personal list, or by using the DSLIST Settings panel of the data set list facility (ISPF option 3.4).

ISPF does not automatically store workstation file names or data set name levels in the data set reference list, though you can add them manually.

Reference Library List

The reference library list is a special personal library list named REFLIST that contains a list of up to 30 library names that you have referenced through panels or ISPF services. The reference library list is updated by the system when ISPF uses ISPF libraries. In all other respects, it functions like a regular personal library list.

How to Get a List of Your Personal Lists

To see a list of your personal lists, do one of the following:

- Select the **List of Personal Data Set Lists** or the **List of Personal Library Lists** choice from the RefList pull-down on the Edit Entry, View Entry, or Utilities panel action bar.
- Select the **Open List of Lists** choice from the File pull-down on the Personal Data Set List or Personal Library List panel action bar.
- Type MORE on the command line of the Personal Data Set List panel or the Personal Library List panel.
- Enter the REFOPEND (for data set lists) or REFOPENL (for library lists) on any ISPF command line.

How to Create a Personal List

There are several ways to create personal lists:

- Type data set names into an empty list.
  1. Select the **New** choice from the File pull-down on the Personal Data Set List panel action bar or use the NEW primary command to display an empty temporary list.
  2. Type in the data set names.
  3. Save the list.
- Use the reference data set list as a starting point.
  1. Display the reference data set list by selecting the **Reference Data Set List** choice from the RefList pull-down on the View Entry, Edit Entry, or Utilities panel action bar.
  2. If you want to save this entire list as a personal data set list, select the **Save as...** choice from the File pull-down on the Reference Data Set List panel action bar.
  3. If you want to save some subset of the reference data set list, see the next item for more information.
- Use an existing list as a base, edit it, and save it.
  1. Display an existing list.
  2. Modify this list by typing over data set names or adding new ones. Do not *select* any lists.
  3. Select the **Save** choice from the File pull-down on the Personal Data Set List panel action bar.
- Use an existing list as a base and save it under another name.
  1. Display an existing list.
  2. Modify this list by typing over data set names or adding new ones. Do not select a choice.
  3. Select the **Save as...** choice from the File pull-down on the Personal Data Set List panel action bar and assign a unique name to this list.
  4. ISPF displays the new list; the original list remains unchanged if **Auto save** is set to Off.

Retrieving Data From Personal Lists

You have three choices for retrieving data from personal lists.

- Use the NRETRIEV function key. Assign a function key to the value ***NRETRIEV***. On panels where NRETRIEV is available (such as edit, view, and some of the utilities), pressing the NRETRIEV key fills in the library name fields, data set

name field, or workstation name field based on where the cursor is when the key is pressed. See "Command Interface (Fast Path) to the Personal List Function" on page 80 for more information.

- Use the RefList pull-down on the View Entry, Edit Entry, Utilities, or other panels that have the RefList pull-down available. From within these options, you can point to a data set or library name and press enter to retrieve or use the name you selected. See "Personal List Modes" on page 68 for more information.
- Use the command interface (fast path commands) described in "Command Interface (Fast Path) to the Personal List Function" on page 80.

Using Personal Lists to Create Customized DSLIST Displays

You can use any personal data set list or personal library list to create a data set list (similar to ISPF option 3.4) which contains multiple level names. The easiest way to use a Personal Data Set List to create a DSLIST is to type

`DSLIST listname`

on an ISPF command line. *Listname* is the name of the personal data set list. If you are already viewing a DSLIST, you can add names listed in a personal data set list by typing

`APPEND listname`

on the command line.

You can also crate a list with the **L** action while displaying a list or a list of lists.

Personal Data Set List Panel

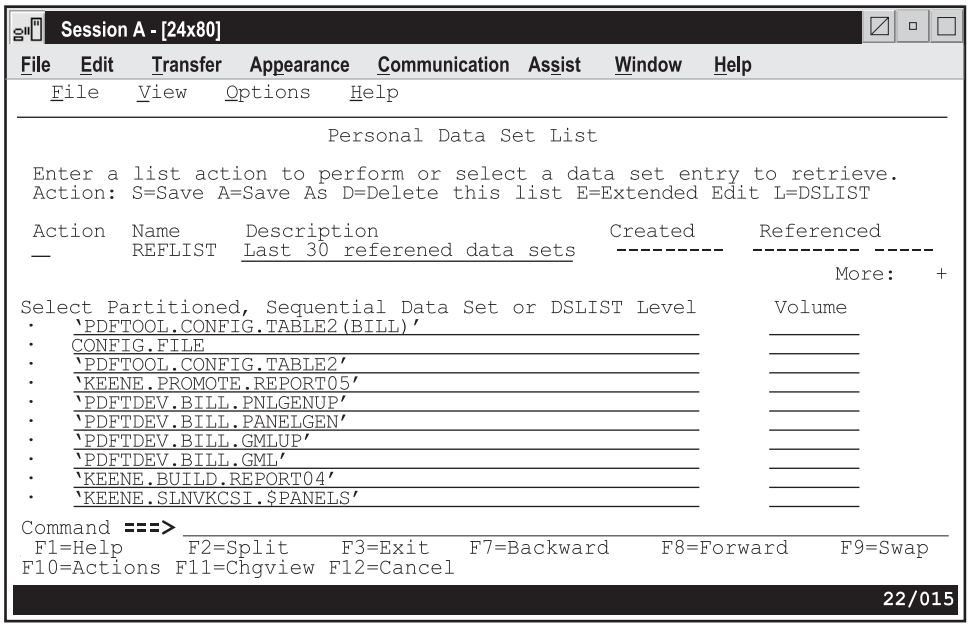


Figure 32. Personal Data Set List Panel (ISRPDSNL)

You can work with your personal data set lists in three ways:

- Use the choices in the File, View, or Options pull-downs.
- Select one of the point-and-shoot options (for example, Save As).

- Type an action mnemonic in the **Action** field and press Enter. Actions are listed at the top of the panel.

Note: The commands and actions on the Personal Data Set List panel are similar to those on the Personal Library List panel. The descriptions here apply to both panels, although the Library List panel does not contain the Sort options.

There are two commands you can use on this panel:

MORE
Displays the list of all of your personal data set lists. This is the same action as selecting *Open List of Lists* from the **File** pull-down.

NEW Saves the current list and displays a new empty list. This is the same action as selecting *New List* from the file pull-down.

Personal Data Set List Panel Action Bar Choices

The Personal Data Set List panel action bar choices function as follows:

File The File pull-down offers you the following choices:

1 **New List** Displays an empty (temporary) personal data set list. After you save this list, it is permanent until you delete it.

2 **Open List of Lists** Displays a list of your personal data set lists. You can open a list to change it and make it the current active list.

3 **Save** Saves the current contents of a personal data set list.

4 **Save as...** Saves the current contents to a new personal data set list. You are prompted for a list name and optional description.

5 **Delete** Deletes the current personal data set list. You are not asked to confirm the delete action. After the current list is deleted, ISPF displays an empty personal data set list as if you had requested a **New** action.

6 **Edit** Enters the personal list edit dialog.

7 **DSLIST** Builds a DSLIST based on list entries.

8 **Cancel** Cancels the function.

9 **Exit** Returns you to the panel from which you accessed the personal list.

View The View pull-down offers you the following choices:

1 **Show data sets** The list contains the data set entries.

2 **Show data sets and notes** The list contains the data set entries with notes.

3 **Sort by data set name** The data set list is sorted by **data set name** field. The sort routine deletes duplicate names, sorts blank entries to the end of the list, saves the list, and redisplays the personal list.

Note: If a volume or entry description exists without a data set field entry, the volume and description are deleted by the sort routine.

4 **Sort by data set volume** The data set list is sorted by data set

volume field. The sort routine deletes duplicate names, sorts blank entries to the end of the list, saves the list, and redisplays the personal list.

Note: If a volume or entry description exists without a data set field entry, the volume and description are deleted by the sort routine.

5 **Sort by data set note** The data set list is sorted by data set note field. The sort routine deletes duplicate names, sorts blank entries to the end of the list, saves the list, and redisplays the personal list.

Note: If a volume or entry description exists without a data set field entry, the volume and description are deleted by the sort routine.

Options

The Options pull-down offers you the following choices:

1 **Personal List Settings** Displays the Personal List Settings panel (ISRRFOPT). From this panel, you can alter all settings that affect personal lists.

2 **Browse shared lists**

Help The Help pull-down offers you the following choices:

1 Tutorial index...
2 Appendices
3 Index

Personal Data Set List Panel Fields

The fields on the Personal Data Set List Panel function as follows:

Action

The following choices are valid in the Action field:

Note: The dots in the **Action** field are point-and-shoot selectable. If you select a data set or library name, and you invoked this dialog from a panel that supports the RefList pull-down, ISPF retrieves the selected data set or library name, terminates this panel, and places the name you selected in the ISPF Library or Other Data Set Name field. If you have the RefMode set to *List Execute*, ISPF also simulates pressing the Enter key on the panel.

S=Save

Saves the current list. If the list is new, you are prompted to enter a name for the list.

A=Save as

Saves the current list with a different list name.

D=Delete this list

Deletes the personal data set list that you are working with. You are asked to confirm this delete action. The currently active list cannot be deleted.

E=Extended edit

Invokes the personal list editor dialog. This enables you to insert, repeat, and delete lines in the list.

L=DSLIST

Invokes DSLIST based on list entries.

Name The name of the personal data set list.

Description
A brief description of the personal data set list. The **Description** field is an input field. You can change the description for all personal lists except the reference list (REFLIST).

Created
The date the personal data set list was created.

Referenced
The last date/time the personal data set list was referenced.

Note: Create and last referenced dates are not maintained for reference lists (the lists named REFLIST).

Personal Data Set Lists Panel

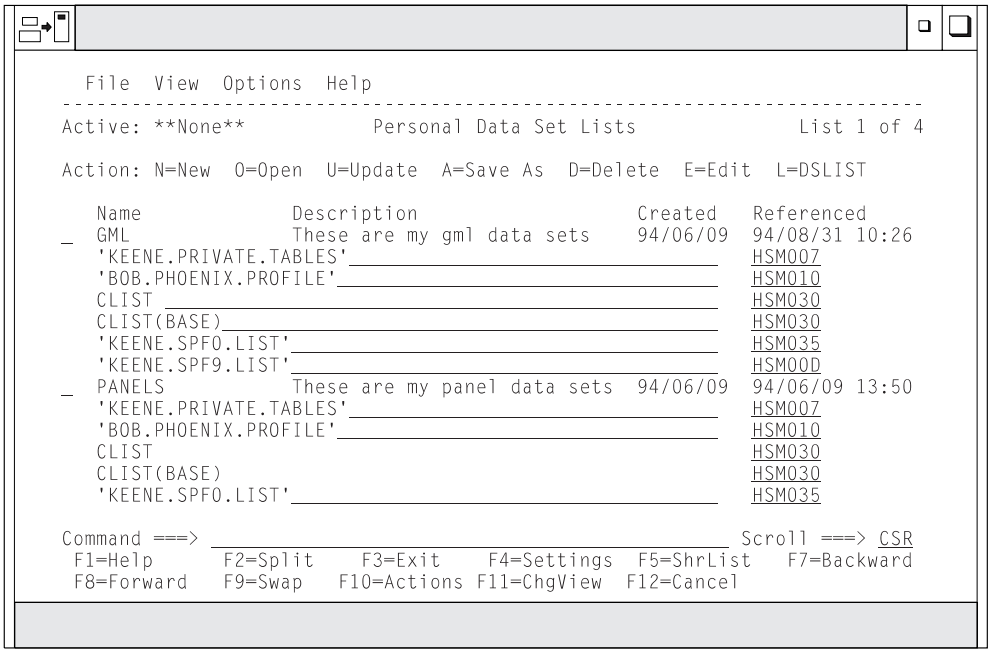


Figure 33. Personal Data Set Lists Panel (ISRPLTAB)

The Personal Data Set Lists panel shows a list of your personal data set lists. You can show the Personal Data Set Lists panel by selecting it from the RefList pull-down or by using the REFOPEND command.

The Personal Library Lists panel is similar. You can display a list of personal library lists by selecting the choice from the RefList pull-down or by using the REFOPENL command.

Personal Data Set Lists Panel Action Bar Choices

The Personal Data Set Lists Panel action bar choices function as follows:

Note: The current setting is shown as an unavailable choice; that is, it displays in blue (the default) with an asterisk as the first digit of the selection number. If you are running in GUI mode, the choice is *grayed*.

File The File pull-down offers you the following choices:

1 **New List** Displays a new personal list.

2 **Open** Displays the personal list entries.

3 **Save as...** Saves the current list entries to a new list.

4 **Delete...** Deletes the list.

5 **Edit** Invokes the personal list edit dialog.

6 **DSLIST** Invokes DSLIST based on list entries.

7 **Exit** Returns you to the panel from which you accessed the Open dialog.

View The View pull-down offers you the following choices:

1 **Standard View** Displays a list of personal lists with list name, list description, and list statistics.

2 **Extended View** Displays a list of personal lists with list name, list description, list statistics, and a partial view of list entries.

3 **Sort by name** Sorts the displayed list alphabetically by the **Name** field.

4 **Sort by description** Sorts the displayed list alphabetically by the **Description** field.

5 **Sort by created** Sorts the displayed list in descending order by the **Created** field.

6 **Sort by referenced** Sorts the displayed list in descending order by the **Referenced** field.

Options

The Options pull-down offers you the following choices:

1 **Personal List Settings...** Displays the Personal List Settings panel, from which you can alter all settings that affect personal lists.

2 **Browse shared lists...** Displays shared personal data set lists. See "Shared Personal Lists" on page 78 for additional information.

Help The Help pull-down offers you the following choices:

1 Tutorial index...
2 Appendices
3 Index

Personal Data Set Lists Panel Fields

The fields on the Personal Data Set Lists Panel function as follows:

Action

The following choices are valid in the Action field:

Note: The dots in the **Action** field are point-and-shoot selectable. If you are in GUI mode and select the action field, ISPF displays the Personal List Commands pop-up so that you can select the action you want to take. In 3270 mode, selecting a list opens the list. This means that you can open a list by moving the cursor to the action field and pressing Enter.

N=New

Displays an empty (temporary) personal data set list. Once you save this list, it is permanent until you delete it.

O=Open
Opens the selected list to modify it, perform actions, or selections of data sets.

A=Save as
Saves the current contents of the selected list to a personal data set list. You will be prompted for a list name and optional description.

D=Delete
Deletes the selected personal data set list. You will be asked to confirm the delete action. The currently active list cannot be deleted.

E=Edit
Invokes the personal list editor dialog.

L=DSLIST
Invokes DSLIST based on list entries.

Name The name of the personal data set list. The **Name** field is a point-and-shoot sort field. The current list is indicated by two asterisks (**) to the right of the name.

Description
A brief description of the personal data set list.

Created
The date the personal data set list was created. The **Created** field is a point-and-shoot sort field.

Referenced
The last date/time the personal data set list was referenced. The **Referenced** field is a point-and-shoot sort field.

Note: A personal list is updated whenever a save action is performed against it.

Note: The LOCATE command is supported as described below:
L *xxxxxxxx*
LOC *xxxxxxxx*
LOCATE *xxxxxxxx*

where: *xxxxxxxx* is the name of the list you wish to locate.

An asterisk is supported in the last position of the list name; for example, you can enter `LOCATE PRIV*` to locate the list named PRIVATE.

The SELECT command is supported as described below:
S *nnnnnnnn A*
SEL *nnnnnnnn A*
SELECT *nnnnnnnn A*

where: *nnnnnnnn* is the name of the list you want to select and A is the action to perform.

An asterisk is supported in the last position of the list name; for example, you can enter `SELECT PRIV* L` to select the list named PRIVATE, with an action of *L* for the DSLIST action.

Shared Personal Lists

Personal lists (library and data set) can be shared with other users on the system. Tables ISRPLSHR for data sets and ISRLLSHR for libraries are shared lists. They are kept in an ISPTLIB concatenated data set. Private lists are tables ISRPLIST (for data set lists) and ISRLLIST (for library lists). They are kept in the ISPPROF user profile data set.

Shared lists are created by using ISPF option 3.3 to copy an existing personal list table from a user profile data set to a data set concatenated to ISPTLIB. You must rename the table to ISRPLSHR (for data set lists) or ISRLLSHR (for library lists) during the copy operation. You can open a shared list and use it; you cannot update or delete a shared list. You can also save it to a private list using the Save As function available from the Personal Data Set Lists panel (ISRPLTAB) and the Personal Library Lists panel (ISRLLTAB). You must save the shared list to a personal list before you can retrieve names from it.

Use the Options pull-down on these two panels to view the shared lists and to save them locally. When you specify shared lists, the following actions are available:

- Open (to interact with the list in any supported way)
- Save As (to save the contents of the current list to a new list)

Name Retrieval with the NRETRIEV command

The ISPF command table contains an entry named NRETRIEV. On enabled panels (such as edit and browse), NRETRIEV retrieves the library names from the current library referral list, or data set or workstation file names from the current data set referral list. The user is responsible for assigning the NRETRIEV command to a PF key.

When the cursor is **not** in the *Other Data Set Name* field, the *volume* field, or the *workstation file name* field, and the NRETRIEV key is pressed, the ISPF library fields are filled in from the current list. As long as the cursor is not placed in these fields, subsequent presses of the NRETRIEV key will retrieve the next library concatenation from the list.

When the cursor **is** in the *Other Data Set Name* field, the *volume* field, or the *workstation file name* field, and the NRETRIEV key is pressed, the other data set name or workstation name field is filled in from the current data set list. ISPF attempts to determine if the name in the list is a workstation or data set name. As long as the cursor is placed in these fields, subsequent presses of the NRETRIEV key will retrieve the next data set or workstation name from the list.

Use the personal list settings panel to force the NRETRIEV command to verify the existence of a data set before retrieving it. If verification is active, then a check is made to see if a data set name exists before a retrieval attempt. If a volume name is not in the personal list entry, then the catalog is checked to see if the data set name is cataloged. If a volume name exists, an OBTAIN macro is used to check the volume for the data set. Verification does not check ISPF library names or workstation names, and does not check for the existence of PDS(E) members. In the data set list *dsname level* field, verification is inactive and workstation names are never retrieved.

NRETRIEV is enabled on the following options:

- View, including extended move, copy, create, and replace panels
- Edit, including extended move, copy, create, and replace panels
- Library Utility (Option 3.1)
- Data Set Utility (Option 3.2)
- Move/Copy Utility (Option 3.3)
- Data Set List (Option 3.4)
- Reset ISPF Statistics (Option 3.5)
- Hardcopy Utility (Option 3.6)
- Workstation Transfer (Option 3.7.2)
- SuperC (Options 3.12, old and new, and Option 3.14)
- SCLM Options:
  - View (Option 1)
  - Edit (Option 2)
  - Member list (Option 3.1)
  - Migration (Option 3.3)
  - Build (Option 4)
  - Promote (Option 5)

SCLM Considerations for NRETRIEV

The NRETRIEV command is enabled to work in several of the SCLM options. There are certain restrictions and considerations to keep in mind when you choose to use NRETRIEV in SCLM.

SCLM Restrictions

- The NRETRIEV key within SCLM does **not** use the standard reference list or personal lists. Instead, it uses a ***stack*** that is stored internally. The stack is not editable. The stack is saved from session to session as a single-line table called ISRSLIST.

  NOTE: In the SCLM View option, the ***other data set name*** field **does** use the standard reference list because the *other data set name* field has no particular meaning to SCLM.
- In SCLM, there is no validation of saved or retrieved names. That means that if you type in a library name and press enter, it is added to the list of saved names, even if SCLM does not process it. This contrasts with the standard reference list processing, which does not add a data set or library name until the data set or library is successfully allocated.
- On name retrieval (when the NRETRIEV key is pressed) there is no validation of the existence of data sets or libraries.
- The regular NRETRIEV command is screen independent (it uses a separate list indicator for each screen in split screen mode). There is only 1 position locator for SCLM lists. This means that split screens with SCLM NRETRIEV will use the same pointer into the list. An NRETRIEV on screen 1 followed by an NRETRIEV on screen 2 will get list entries 1 and 2 respectively.

Stack Management for SCLM

A library name (or concatenation) is added to the *saved library* list by pressing enter on a panel that supports saving names. If the library or concatenation exists in the list already, it is moved to the top of the list. Where the ***project*** field, or the first ***group*** field is an output field (SCLM options 2, 3, 4, and 5), the output fields are not used in the comparison between what was typed on the panel and what is already in the list. This enables you to work in different but similar projects.

In other words, on the edit screen that has both the **project** and **group1** as output fields, the concatenation:

```
SCLM Library:
   Project...: PDFTDEV
   Group ....: DGN      ....STG    ....INT    ....SVT
   Type .....: ARCHDEF
   Member ...:
```

would match

```
SCLM Library:
   Project...: PDFTOS25
   Group ....: JSM      ....STG    ....INT    ....SVT
   Type .....: ARCHDEF
   Member ...:
```

Similarly, where groups 2, 3, and 4 are not used, those groups are not used when checking to see if the name already exists.

If a match is found, the existing entry in the list is moved to the top of the list.

Command Interface (Fast Path) to the Personal List Function

You can use the following commands to access the referral list function rather than using the action bar pull-down menus:

REFLISTD xx
Start the personal data set list dialog with the reference data set list and retrieve the data set in position *xx*. The *xx* parameter is optional. This sets the current data set list to the reference list (REFLIST).

REFLISTL xx
Start the personal library list dialog with the reference library list and retrieve the data set in position *xx*. The *xx* parameter is optional. This sets the current library list to the reference list (REFLIST).

REFACTD nnnnnnnn xx
Start the personal data set list named *nnnnnnnn* and retrieve the data set in position *xx*. For example, enter

```
REFACTD MYLIST 2
```

to retrieve the second data set from the personal data set list named MYLIST and place it in the **Data Set Name** field. The *nnnnnnnn* and *xx* parameters are optional.

REFACTL nnnnnnnn xx
Start the personal library list named *nnnnnnnn* and retrieve the library in position *xx*. For example, enter

```
REFACTL MYLIB 1
```

to retrieve the first library from the personal library list named MYLIB and place it in the **Library** field. The *nnnnnnnn* and *xx* parameters are optional.

REFOPEND
Start the personal data set open dialog.

REFOPENL
Start the personal library open dialog.

REFADDD nnnnnnnn

Update the personal data set list named *nnnnnnnn* with the most recently referenced data set. For example, enter

```
REFADDD NEWLIST
```

to add the most recently referenced data set to the personal data set list named NEWLIST.

REFADDL nnnnnnnn

Update the personal library list named *nnnnnnnn* with the most recently referenced library. For example, enter

```
REFADDL NEWLIB
```

to add the most recently referenced library to the personal library list named NEWLIB.

NRETRIEV

Retrieve a name from the current library or data set list on panels which support NRETRIEV commands. This command is normally assigned to a program function (PF) key. NRETRIEV uses the position of the cursor to determine what type of personal list to use and what fields on the panel to fill in. See "Name Retrieval with the NRETRIEV command" on page 78 for more information about NRETRIEV.

Using Function Keys with Personal Lists

You can set function keys to process any of the fast path commands, as shown in Figure 34.

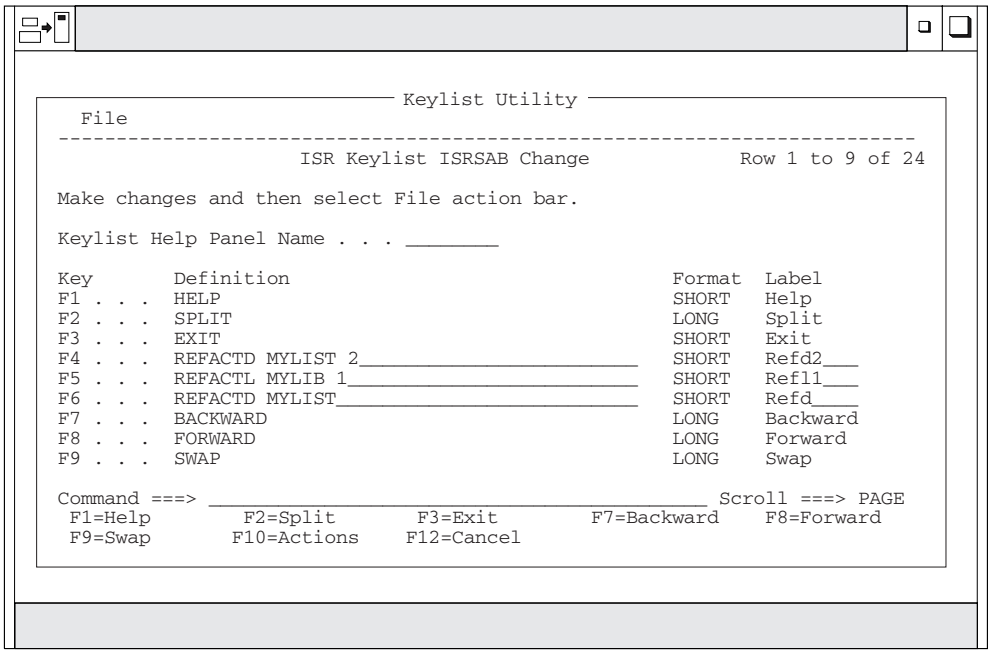


Figure 34. Defining Function Keys to Issue Referral List Commands

If you used these function key settings, you could do the following:

- Press F4 to place the second data set name on the personal data set list named MYLIST in the Data Set Name field.

- Press F5 to place the first library name on the personal library list named MYLIB in the Library field.
- Press F6 to display the personal data set list named MYLIST so that you can select a data set to process.

  Note: You could also type a number on the command line and press F6 to place the data set name in the specified position on the personal data set list named MYLIST in the Data Set Name field; for example, if you type a 6 on the command line and press F6, the sixth data set on MYLIST would be placed in the Data Set Name field.

Example of an ISPF Panel That Uses a Referral List

Figure 35 shows the panel definition for a panel that uses a referral list.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)
 0B TYPE(AB)
 04 TYPE(ABSL)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 0C TYPE(NT) SKIP(ON)
 11 TYPE(SAC)
 12 TYPE(CEF) PADC(USER)
 13 TYPE(NEF) PADC(USER)
 19 TYPE(DT)
 22 TYPE(WASL) SKIP(ON)
 08 TYPE(CH)
 10 TYPE(ET)
)ABC DESC('RefList')
 PDC DESC('Reference Data Set List')
   ACTION RUN(ISRRLIST) PARM('RL1')
 PDC DESC('Reference Library List')
   ACTION RUN(ISRRLIST) PARM('LR1')
 PDC DESC('Personal Data Set List')
   ACTION RUN(ISRRLIST) PARM('PL1')
 PDC DESC('Personal Data Set List Open')
   ACTION RUN(ISRRLIST) PARM('PL2')
 PDC DESC('Personal Library List')
   ACTION RUN(ISRRLIST) PARM('LL1')
 PDC DESC('Personal Library List Open')
   ACTION RUN(ISRRLIST) PARM('LL2')
```

Figure 35. Example Panel Definition Using a Referral List (Part 1 of 3)

```
)ABCINIT
   .ZVARS=REFLIST
)ABC DESC('RefMode')
 PDC DESC('List Execute') UNAVAIL(ZRME1)
   ACTION RUN(ISRRLIST) PARM('EEX')
 PDC DESC('List Retrieve') UNAVAIL(ZRME2)
   ACTION RUN(ISRRLIST) PARM('ERT')
)ABCINIT
   .ZVARS=LISTFILE
  VGET (ZELIST) PROFILE
  IF (&ZELIST = 'EXECUTE')
    &zrme1 = 1
    &zrme2 = 0
    &listfile = 2
  ELSE
    &zrme1 = 0
    &zrme2 = 1
    &listfile = 1
)BODY  CMD(ZCMD)
   ⋮

)INIT
   ⋮

)REINIT
   REFRESH (PRJ1,LIB1,LIB2,LIB3,LIB4,TYP1,MEM,DSN) /*refresh panel vars*/
)PROC
  /* the following is the logic for reference or personal data set list*/
 VGET (ZRDSN) SHARED         /* get data set reflist key var     */
 IF (&ZRDSN ^= ' ')          /* if reflist has set dsname var    */
   &DSN = &ZRDSN             /* set panel other dsname to zrdsn  */
   VGET (ZREFVOLM) PROFILE   /* get volume retrieve mode         */
   IF (&ZREFVOLM = 'ON')     /* if volume retrieve on            */
     &VOL = &ZRVOL           /* set panel volume to zrvol        */
   &ZRDSN = ' '              /* blank zrdsn                      */
   &ZRVOL = ' '              /* blank zrvol                      */
   VPUT (ZRDSN ZRVOL) SHARED /* return blank reflist vars        */
   .CURSOR = DSN             /* set cursor to panel dsname field */
   VGET (ZELIST) PROFILE     /* get edit execute/retrieve mode   */
   IF (&ZELIST ^= 'EXECUTE') /* determine if retrieve or execute */
     .MSG = ISRDS003         /* force redisplay if retrieve mode */
  /* End of logic for reference or personal data set list        */
  /* the following is the logic for reference or personal library list */
 VGET (DSALSEL) SHARED       /* get library reflist key var      */
 IF (&DSALSEL ^= ' ')        /* if reflist has set lib indicator */
   VGET (DSA1,DSA2,DSA3,DSA4,DSA5,DSA6,DSA7) SHARED /* get vars  */
   &PRJ1 = &DSA1             /* set panel project                */
   &LIB1 = &DSA2             /* set panel library 1              */
   &LIB2 = &DSA3             /* set panel library 2              */
   &LIB3 = &DSA4             /* set panel library 3              */
   &LIB4 = &DSA5             /* set panel library 4              */
   &TYP1 = &DSA6             /* set panel type                   */
   &MEM  = &DSA7             /* set panel member                 */
```

Figure 35. Example Panel Definition Using a Referral List (Part 2 of 3)

```
 &DSN = ' '                /* blank panel other dsname         */
   &DSALSEL = ' '            /* blank reflist lib indicator      */
   VPUT (DSALSEL) SHARED     /* return to shared pool            */
   .CURSOR = MEM             /* set cursor to panel member field */
   VGET (ZELIST) PROFILE     /* get edit execute/retrieve ind    */
   IF (&ZELIST ^= 'EXECUTE') /* determine if retrieve or execute */
     .MSG = ISRDS003         /* setmsg if retrieve mode          */
  /* End of logic for reference or personal library list         */
)END
```

Figure 35. Example Panel Definition Using a Referral List (Part 3 of 3)

Example of an ISPF Panel That Enables NRETRIEV

Figure 36 on page 85 shows the panel definition for a panel that uses a referral list.

```
)BODY
%---------------------------  NRETRIEV Test Panel  ---------------------------%
%COMMAND%===>_ZCMD
              +
+
+  Project ===>_PROJECT +
+  Group   ===>_GROUP1  +===>_GROUP2  +===>_GROUP3  +===>_GROUP4  +
+  Type    ===>_TYPE    +
+  Member  ===>_MEMBER  +

+  DS Name ===>_OTHERDSN                                  +
+  Volume  ===>_VOLUME+

+  Workstation File ===>_WSFILE                            +
)INIT
 .NRET = ON                       /* Make NRETRIEV key active*/
)REINIT
   REFRESH (*)
   .NRET = ON                     /* Make NRETRIEV key active*/
)PROC
.NRET = OFF          /*IMPORTANT - Make NRETREIV key inactive*/
VGET (ZVERB) SHARED
 IF (&ZVERB = NRETRIEV)    /* if NRETRIEV was entered */
  IF (.CURSOR NE OTHERDSN, VOLUME, WSFILE)
    .NRET = LIB                    /* Reset data set counter*/
    IF (&ZNRLIB = YES)          /* If library retrieve was OK*/
      .CURSOR = MEMBER          /* set cursor to member field*/
      &PROJECT  = &ZNRPROJ       /* set library variables from*/
      &GROUP1   = &ZNRGRP1       /* the variables set by      */
      &GROUP2   = &ZNRGRP2       /*  NRETRIEV                 */
      &GROUP3   = &ZNRGRP3
      &GROUP4   = &ZNRGRP4
      &TYPE     = &ZNRTYPE
      &MEMBER   = &ZNRMEM
      &OTHERDSN = &Z             /* Blank out odsn field      */
      &VOLUME   = &Z             /* Blank out volume field    */
      &WSFILE   = &Z             /* Blank out workstation file*/
      .MSG = ISRDS013            /* Indicate good retrieval   */
    ELSE .MSG = ISRDS011         /* Else bad library list     */
   ELSE
    .NRET = DSN
    IF (&ZNRDS = YES)            /* If dsname retrieve was OK */
      IF (&ZNRWSN = H)           /* If ISPF thinks it's host  */
        .CURSOR = OTHERDSN       /* Move cursor to dsn name   */
        &OTHERDSN = &ZNRODSN     /* Set other dsn name        */
        &VOL      = &ZNRVOL      /* Set volume variable       */
        &WSFILE   = &Z           /* Blank out workstation file*/
      ELSE                       /* ISPF thinks it's a wsfile */
        .CURSOR = WSFILE         /* Set cursor to ws file     */
        &WSFILE   = &ZNRODSN     /* Set ws file name variable */
        &VOL      = &Z           /* Blank volume and data set */
        &OTHERDSN = &Z           /* To avoid visual confusion */
      .MSG = ISRDS014            /* Indicate good retrieval   */
    ELSE .MSG = ISRDS012         /* Else bad ds referral list */
)END
```

Figure 36. Example Panel Definition Enabling NRETRIEV

Chapter 4. Using Commands, Function Keys, and Light Pen or Cursor Selection

This chapter explains how to use the ISPF system commands, the function keys and their default assignments, and the light pen and cursor-select facilities.

You can use commands to request processing functions. These are the levels of commands:

System commands
Provided by the DM component and always available to a user, unless explicitly overridden by an application, a user, or a site.

User or Site commands
Defined by the site administrator (in the ISPF Configuration table) and available to a user, in addition to the system commands.

Application commands
Available to a user throughout the processing of an application.

Function commands
Meaningful only while using a particular function within an application.

System, user, site, and application commands are defined by using command tables. The DM component processes these commands. System, user, site, and application command processing is generally transparent to the dialog functions. For example, HELP is a system command.

Function commands include all commands that are processed by a dialog function. For example, the NUMBER command within the ISPF Editor (option 2) is a function command.

You can enter a command by:

- Typing the information on the command line, or in the command field, and pressing the Enter key. This includes the command field in View, Browse, Edit, and Table Displays, as well as the command field on a panel.
- Pressing the function key set to the desired command.
- Selecting an **Attention** field by using the light pen or the cursor-select key. The cursor-select key is a hardware feature on 3179, 3179G, 3180, 3278, 3279, and 3290 terminals.

ISPF intercepts all commands, regardless of whether the command is typed in the command field or entered with a function key, light pen, or cursor-select key. The DM component performs the command if it matches an entry in the application, user, or system command table. Otherwise, it is assumed to be a function command and is passed to the dialog function.

You can pass commands to the operating system by entering the appropriate ISPF-provided command (TSO) followed by the actual TSO command. For example:

```
===> TSO LISTC LEVEL(Z77PHJ)
```

You can stack commands to be run by entering a special delimiter between the commands. For example, entering:

```
===> UPDATE BLDG DEPT NAME; MENU ABC
```

causes the UPDATE command to run first. When it completes, the MENU command starts. The default delimiter is a semicolon (;), which you can change with the ISPF SETTINGS option (see the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II*).

Commands cannot be stacked following the:

- HELP command. HELP processing deletes any remaining commands in the stack.
- RETRIEVE command.

ISPF System Commands

Table 3 describes the ISPF system commands in alphabetical order.

Table 3. ISPF System Commands

Command	Description
ACTIONS	Moves the cursor between the action bar and the panel body.
AUTOTYPE	Allows you to type a partial data set or member name, press a program function key, and ISPF completes the name for you. **Note:** AUTOTYPE is not a true ISPF system command because it is not built into the base code of ISPF and it works only on panels that are written to understand it.
BACKWARD	Alias for the UP command. Scrolls toward the top of the data.
BOTTOM	Alias for the DOWN MAX command. Scrolls to the bottom of the data.
CANCEL	If CANCEL is requested from an action bar pull-down, the action taken depends on whether you are running in host or GUI mode: • In host mode, the pull-down is removed and the cursor is positioned on the first action bar choice. • In GUI mode, the CANCEL command is returned to the dialog. If CANCEL is requested from a panel displayed using the DISPLAY, TBDISPL, or SELECT service calls, the DM component places the command in ZCMD and sets a return code of 0 from the display screen. If CANCEL is requested from a panel displayed using the DISPLAY or TBDISPL service calls and the panel was defined with a PANEL tag (DTL) or a)PANEL statement, the DM component returns the command in ZVERB and sets a return code of 8 from the display screen.

Table 3. ISPF System Commands (continued)

Command	Description
CMDE	If CMDE is entered on any command line, a pop-up panel (ISPCMDE) with a 234-character command input field is displayed. You can enter up to 234 characters using the entry field provided. ISPF allows TSO commands, CLISTS, and REXX execs and parameters to be entered in the input field. This panel is processed much like the PDF Option 6 panel. Data passed to this panel is translated to uppercase characters. Data passed from this panel remains as it appears on the panel. If input has been entered on the panel from which CMDE was called, it is saved and displayed when the pop-up panel ISPCMDE is displayed.
COLOR	Changes the default colors on seven-color display devices.
CRETRIEV	The actions of the CRETRIEV (conditional retrieve) command are based on the position of the cursor when you enter the command: • If the cursor is within the primary input field when you enter the CRETRIEV command, the command does the same processing as the RETRIEVE command; the DM component places the previous command entered, if any, in the command input field. • If the cursor is not within the primary input field, the CRETRIEV command does the same processing as a CURSOR command; the DM component places the cursor at the beginning of the first input field on the panel, which is usually the option or command field.
CUAATTR	Changes the default values of panel colors, intensities, and highlights for CUA panel element attributes.
CURSOR	Moves the cursor to the first input field on the panel being displayed, generally the option or command field, or moves the cursor to the alternate command field if one has been designated on the)BODY statement. If invoked a second time on a panel with scrollable data, this command causes the cursor to be moved to the second input field. Scrollable data panels include a View, Browse, Edit, or table display panel or a panel with a scrollable dynamic area.
DOWN	Scrolls toward the bottom of the data.
DSLIST [list name | DSname level]	Enables you to build a data set list from any command line. You can specify either a personal data set list name or a data set list name level on the command. If you do not put quotation marks around the dsname level, the TSO prefix is used as the first qualifier in the dsname level. By issuing the command with no parameters, you cause a list of available personal data set lists to be displayed.

ISPF System Commands

Table 3. ISPF System Commands (continued)

Command	Description
DTEST *parameter number*	Enables you to start, or change the conditions of, a dialog test. Specifying a parameter number is required, and different conditions of dialog test result. For example, if you enter DTEST 8 while running an application under Dialog Test, the 7.8 Breakpoints panel is displayed. After setting the breakpoints, you return to your application with the new breakpoints activated. The panels that you can call up with DTEST are: **1** Invoke Functions panel **2** Invoke Display Panel panel **3** Invoke Variables panel **4** Invoke Tables panel **5** Display Browse log panel **6** Invoke Dialog Services panel **7** Invoke Traces selection panel **8** Invoke Breakpoint panel
END	Stops the current operation and returns to the previous menu. If the ISPF Primary Option Menu is displayed, this command ends ISPF. See "Log and List Data Set Processing at the End of a Session" on page 156 for a description of the processing that occurs when the END command is entered from the ISPF Primary Option Menu. When entered on a selection panel displayed by the SELECT service, the END command causes a redisplay of the next higher menu in the hierarchy. When entered on a panel displayed by the tutorial program, it stops the tutorial and causes a redisplay of the menu from which the tutorial was started or the panel from which HELP was requested. When the END command is entered on a panel displayed by a dialog function through the DISPLAY or TBDISPL service, the dialog function must take whatever action is appropriate to terminate and return control. Entry of the END command is signalled by a return code of 8 from the DISPLAY or TBDISPL service.
ENVIRON [ENBLDUMP[ON|OFF]] [TERMTRAC[ON|ERROR|DUMP|OFF]] [TERMSTAT [QUERY]]	Allows you to reduce service time by gathering data that can be helpful in diagnosing problems. Functions provided include: • Enabling Abend dumps when ISPF is not in TEST mode • Tracing and dumping ISPF terminal input and output data and errors • Collecting terminal characteristic information.

Table 3. ISPF System Commands (continued)

Command	Description
EPDF datasetname [Browse] [View] [Macro macroname] [Profile profilename][Panel panelname][Recover] [Format formatname] [Mixed YES|NO]	Enables you to edit, browse, or view a data set from a command line. **Browse** Invoke Browse instead of edit. **View** Use View mode (End/Save/Cancel disabled). **Macro macroname** Invoke the editor or view using the initial macro specified by *macroname*. Not valid with Browse. **Profile profilename** Invoke the editor or view using the edit profile specified by *profilename*. Not valid with Browse. **Panel panelname** Use alternate panel name specified by *panelname*. **Format formatname** Invoke the editor using the format table specified by *formatname*. **Mixed YES|NO** Use mixed option for 5550 terminals.
EXHELP	Provides general information about the contents of a panel.
EXIT	Requests that the current function be ended. When entered on a panel displayed by the tutorial program, EXIT stops the tutorial and causes a redisplay of the menu from which the tutorial was started or the panel from which HELP was requested. • If EXIT is requested from a panel displayed using the DISPLAY, TBDISPL, or SELECT service calls, the DM component returns the command in ZCMD and sets a return code of 0 from the display screen. • If EXIT is requested from a panel displayed using the DISPLAY or TBDISPL service calls and the panel was defined using a PANEL tag (DTL) or a)PANEL statement, the DM component returns the command in ZVERB and sets a return code of 8 from the display screen.

Table 3. ISPF System Commands (continued)

Command	Description
FKA [ON|SHORT|OFF| PREFIX|NOPREFIX]	Toggles through the different forms of the function key area. The first time you enter the FKA command (without parameters), the long form of the function key area is displayed. The long form includes the keys that have a format specified as either long or short in the keylist. If you enter the command again, the short form is displayed. The short form displays only those keys that have the short format specified in the keylist. If you enter the command once again, the keys are removed from the display. Therefore, if you continue to enter the command, the different choices are toggled: • Long form (default) • Short form • No display. The form that you select affects all panels displayed in the session. The DM component updates the system variable ZFKA to represent the current state of the function key area form and saves the value in the system profile. FKA ON displays the long form of the function key area. FKA SHORT displays the short form of the function key area. FKA OFF specifies that the function key area will not be displayed. If you are running in GUI mode, FKA PREFIX displays the F= prefix on the push button. If you are running in GUI mode, FKA NOPREFIX removes the F= prefix from the push button.
FORWARD	Alias for the DOWN command. Scrolls toward the bottom of the data.
HELP	Displays additional information about an error message, or provides tutorial/help information for panels, fields on panels, commands, and options.
INT	Starts the ISPF Workstation Tool Integration dialog. Use the dialog to configure workstation tools and file extensions for ISPF actions and services for Edit and View when used in conjunction with workstation edit and view. See "The ISPF Workstation Tool Integration Program" on page 32 for more information.
ISPDTLC	Invokes the ISPF DTL Conversion Utility. See the *ISPF Dialog Tag Language Guide and Reference* for additional parameters and calling options.
ISPFVAR [LMSG(ON|OFF) | ABTAB(ON|OFF) | PSTAB(ON|OFF) | SESM(ON|OFF) | JUMP(ON|OFF) | EDPRT(ON|OFF) | EURO(ON|OFF) | SPLTLINE(ON|OFF)]	Sets the following system variables: **LMSG** Display long message in pop-up **ABTAB** Tab to action bar choices **PSTAB** Tab to point-and-shoot fields **SESM** Select Session Manager mode **JUMP** Jump from leader dots **EDPRT** Edit the PRINTDS command. **EURO** Enable the EURO currency symbol. **SPLTLINE** Always show split line.

Table 3. ISPF System Commands (continued)

Command	Description
ISPFWORK	Starts the ISPF Workplace.
ISPLIBD [libtype]	Invokes the LIBDEF Display Utility. The optional parameter, libtype, identifies a specific LIBDEF library definition to be displayed. All LIBDEF definitions for the current logical screen are displayed if the parameter is omitted, if the parameter is longer than eight characters, or if the parameter specifies ISPPROF as the library name. See *ISPF Dialog Developer's Guide and Reference* for more information about the ISPLIBD command.
ISPPREP	Allows you to create preprocessed panels, those for which ISPF has partially processed the panel definition before it is stored in the panel data set, either interactively or in batch mode.
ISRRLIST	The action bar interface into referral lists.
ISRROUTE	The action bar interface into the ISPF command stacking routing. ISRROUTE also provides an interface to the SELECT service from the ACTION/RUN statement within a pull-down choice. The parameters are the same as the ISPEXEC interface to the SELECT service.
KEYLIST [PRIVATE|SHARED|OFF|ON]	The parameters on this command determine where, or if, ISPF looks for keylists. The default setting for KEYLIST is equivalent to issuing the KEYLIST PRIVATE command, which means that the program looks in the user's profile table for the keylist specified on a panel before looking in the xxxxKEYS table allocated in ISPTLIB. The KEYLIST SHARED command means that ISPF looks only in the xxxxKEYS table allocated in ISPTLIB for the keylist. Using either the PRIVATE or SHARED parameter performs an implicit KEYLIST ON command. Both of the parameters are local to each application, so setting PRIVATE for application X does not affect application Y, which might be using SHARED. By specifying KEYLIST OFF, you cause ISPF to ignore the keylist on all logical screens and use the ZPF variables for controlling function keys. This is in effect only for the application for which you enter the command. The KEYLIST ON command causes ISPF to recognize keylists again, with the parameter (SHARED or PRIVATE) that was in effect immediately prior to the KEYLIST OFF command. KEYLIST ON and OFF are equivalent to the Enable and Disable keylist choices on the Function keys pull-down discussed in the **Keylist Settings** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II*. SHARED and PRIVATE also appear on the Function keys pull-down in ″Keylist Settings″. The KEYLIST command with no parameters causes the Keylist utility to start.

Table 3. ISPF System Commands (continued)

Command	Description
KEYS	Displays the PF Key Definitions and Labels panel, which allows you to change the ZPF variable settings (ZPFVARs), as in previous versions of ISPF. However, if the KEYS command is issued from a panel with an *active* keylist, the associated Keylist Utility panel Change pop-up window is displayed. **Note:** If the KEYLIST SHARED command has been issued, or the SYSTEM parameter has been specified on the KEYLIST keyword on the)PANEL statement, this action causes only a BROWSE of the keylist. Refer to the *Dialog Developer's Guide and Reference* for more information about the SYSTEM parameter in the)PANEL statement.
KEYSHELP	If KEYSHELP is defined, KEYSHELP provides you with a brief description of each key defined for a panel.
LEFT	Scrolls left.
LIST [PRINT|DELETE|KEEP]	Allows you to process the list data set without exiting ISPF. See "Processing the Log and List Data Sets" on page 153 for a description of using the LIST command.
LOG [PRINT|DELETE|KEEP]	Allows you to process the log data set without exiting ISPF. See "Processing the Log and List Data Sets" on page 153 for a description of using the LOG command.
MSGID [ON|OFF]	With no parameters, displays a message indicating the message ID of the last message displayed. With a parameter of ON or OFF, indicates whether a message number is to be added to the beginning of interactive long message text. During entry to ISPF, the mode is initialized to OFF, and the message ID is not displayed as part of the long message text on interactive displays. If the addition of the message ID would cause long message text to be truncated, the message is displayed in a pop-up window. Messages that have the message number included in the long message text will continue to display the message number, even when MSGID OFF is in effect. Also, the message number will appear twice when MSGID ON is in effect. The MSGID ON/OFF command affects only the current logical screen, so when you are running in split screen, one screen can have MSGID ON and the other MSGID OFF. The MSGID command will return only the MSGID of a message for its own logical screen. An option on the Log Data Set Defaults and List Data Set Defaults panels, which are choices on the Log/List pull-down on the ISPF Settings panel, `Log Message ID . . . _  (/ = Yes)` allows you to select whether the message ID is written to the log data set as part of the long message text. The initial default is unselected. Note that not all lines in the log data set originate from a message member. Therefore, not every line in the log data set will have a message number associated with it. **Note:** This facility does not affect long message text returned by the GETMSG service, messages displayed in the Error Box, or messages displayed by TRACEX.
NOP	The classic *no operation* command.

Table 3. ISPF System Commands (continued)

Command	Description
NRETRIEV	Data set and library name retrieved. See "Name Retrieval with the NRETRIEV command" on page 78 for more information.
PANELID [ON|OFF]	Indicates whether or not the panel identifier (ID) is to be displayed. If you enter PANELID without any parameters, the command toggles the display of the panel ID. If an action bar is present, the ID is displayed immediately below the action bar, indented one space. If an action bar is not present, the DM component displays the ID indented one space in the left corner of line 1 on the panel. **Note:** In GUI mode, the panel identifier is placed on the title bar rather than on the first line of the panel. During initial entry to ISPF, the PANELID is set to OFF. The ID is displayed only if the panel contains a protected-field attribute byte in row 1 column 1 (relative to the action bar) and is padded with one blank.
PFSHOW [ON |OFF|TAILOR]	Toggles through the different forms of the function key area. The first time you enter the PFSHOW command (without parameters), the long form of the function key area is displayed. If you enter the command again, the short form is displayed. If you enter the command once again, the keys are removed from the display. Therefore, if you continue to enter the command, the different choices are toggled: • Long form (default) • Short form • No display. The form that you select affects all panels displayed in the session. The DM component updates the system variable ZPFSHOW to represent the current state of the function key area form and saves the value in the system profile. PFSHOW ON displays the long form of the function key area. PFSHOW OFF specifies that the function key area will not be displayed. PFSHOW TAILOR displays a panel that lets you specify the set of function keys (primary, alternate, or all) for which definitions are to be displayed and the number of keys per line to display in each function key definition line.

Table 3. ISPF System Commands (continued)

Command	Description
PRINT	Records a snapshot of the physical screen image in the list data set for subsequent printing. For the PRINT, PRINT-HI, PRINTL, and PRINTLHI commands, a screen image can exceed 121 characters. When it does, the line must be split when the output is being directed to a printer other than a 3800. The line length is obtained from a user-modifiable specification on the ISPF Settings panel for the list data set. The default length for printing is 121 characters. **Using Print Commands with DBCS** The print commands are affected in the DBCS environment as follows: • DBCS character printing Because shift-out and shift-in characters do not occupy positions on a printer, ISPF inserts a blank character before each shift-out and after each shift-in. • Fields affected by the OUTLINE keyword Field-outlining information is imbedded in the record as a set-attribute (SA) order. Each SA order occupies three bytes. An SA is required to start field-outlining, one to end field-outlining, and one to change field-outlining. Therefore, each affected field normally takes six additional bytes. Thus, the record-length of print command output is larger than the screen width. The LIST file should be large enough to contain the expanded records. If not, the output might not print correctly. **Note:** The PRINT command is disabled if you are running in GUI mode.

Table 3. ISPF System Commands (continued)

Command	Description
PRINTG	Allows you to send the information on the current logical screen to a Graphical Data Display Manager (GDDM*) graphics printer. **Notes:** 1. In split-screen mode, ISPF adds the split line to the top logical screen. If you issue the PRINTG command from the top screen, the split line is printed along with the logical screen. 2. Also, in split-screen mode, PRINTG prints all data in the visible portion of the logical screen, but only the graphics area data in the nonvisible portion of the logical screen. If you use the other print commands (PRINT, PRINT-HI, PRINTL, and PRINTLHI) to print screen images containing a graphics area, the part of the screen containing the graphics area prints as blanks. If you issue the PRINTG command as a COMMAND option on a DISPLAY service request, only data already defined to GDDM at the time the service request is issued will be printed. Any GDDM fields defined by the dialog (using GDDM commands) prior to issuing the DISPLAY service request will be printed. Before issuing the PRINTG command from a command line you must first have initialized the GDDM graphic interface using the GRINIT service. PRINTG does not provide return codes to a dialog; however, it does display completion or error messages. For information about how to specify parameters related to using the PRINTG command, see the **Print Graphics Parms** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II*. **Note:** The PRINTG command is disabled if you are running in GUI mode.
PRINT-HI	Same as PRINT, except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. See *Using Print Commands with DBCS* under the PRINT command. **Note:** The PRINT-HI command is disabled if you are running in GUI mode.
PRINTL	Causes a *snapshot* of the logical screen image to be recorded in the ISPF list file for subsequent printing. In split-screen mode, the PRINTL command prints what would be seen if split-screen were not in effect. See *Using Print Commands with DBCS* under the PRINT command. **Note:** The PRINTL command is disabled if you are running in GUI mode.

Table 3. ISPF System Commands (continued)

Command	Description
PRINTLHI	Same as PRINTL, except that high-intensity characters on the logical screen are printed with overstrikes to simulate the dual-intensity display. See *Using Print Commands with DBCS* under the PRINT command. **Note:** The PRINTLHI command is disabled if you are running in GUI mode.

PSCOLOR	Globally alters the color, intensity, and highlighting of point-and-shoot fields through a pop-up dialog. Valid choices include the following:
	
	`Color Intensity Highlight`
	
	`RED HIGH NONE`
	`PINK LOW BLINK`
	`GREEN REVERSE`
	`YELLOW USCORE`
	`BLUE`
	`TURQ (Turquoise)`
	`WHITE`
	
	To restore the ISPF default values, delete any new values you have entered (leaving the entry fields blank) and press Enter, or select the **Defaults** field.

RCHANGE	Repeats the action of the previous CHANGE command (change one character string to another) (Edit and View only).
REFACTD *nnnnnnnn xx*	Calls the personal data set list named *nnnnnnnn* and retrieves the data set in position *xx*. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFACTL *nnnnnnnn xx*	Calls the personal data set list named *nnnnnnnn* and retrieves the data set in position *xx*. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFADDD *nnnnnnnn*	Updates the personal data set list named *nnnnnnnn* with the most recently referenced data set. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFADDL *nnnnnnnn*	Updates the personal library list named *nnnnnnnn* with the most recently referenced library. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFLISTD *xx*	Calls the reference data set list dialog and retrieves the data set in position *xx*. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFLISTL *xx*	Calls the reference library list dialog and retrieves the library in position *xx*. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFOPEND	Calls the personal data set open dialog. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.
REFOPENL	Calls the personal library list open dialog. See “Command Interface (Fast Path) to the Personal List Function” on page 80 for additional information.

Table 3. ISPF System Commands (continued)

Command	Description
RESIZE	Increases the size of a pop-up window to fill the entire 3270 physical display area. The initial RESIZE command increases the pop-up window to its maximum size, and the following RESIZE reduces the window to its original size.
RETF	Retrieves commands from the command stack moving in the direction from the oldest command in the command stack toward the most recent commands in the command stack. Forward retrieve (RETF) retrieves the oldest command on the command stack, if RETF is entered immediately after a command is executed, before performing a RETRIEVE. See "RETF Command" on page 106 for more information.
RETP	Causes a pop-up panel to be displayed with a list of the last 20 commands in the retrieve stack. Retrieve pop-up (RETP) enables you to select by number the command to be retrieved. The command selected is retrieved to the command line, as it is when using other retrieve commands. You will not be able to change the commands in the retrieve pop-up until the command is selected and retrieved to the command line. The RETP pop-up panel has an OPTIONS action bar choice that allows you to set the minimum number of characters required to save a command in the retrieve stack and to choose whether to position the cursor at the beginning or end of the retrieved command when the command is retrieved to the command line. RETP displays the pop-up panel if the retrieve stack is empty, which allows the user to change the retrieve options. See "RETP Command" on page 106 for more information.
RETRIEVE	Repeatedly entering RETRIEVE causes the commands most recently entered from the primary input field, usually the ZCMD field, to be displayed on the command line. The commands are displayed one at a time, in the reverse sequence to which they were entered (last-in, first-out). This allows you to easily recall a command for resubmission from the command line. You can edit the command before entering it if you wish. See "RETRIEVE Command" on page 105 for more information.
RETURN	Causes an immediate return to a primary option menu or to the display from which you entered a nested dialog. The RETURN command simulates repeated END commands, up to some appropriate stopping point, without displaying intervening panels. See "Using the RETURN Command" on page 107 for more information.
RFIND	Repeats the action of the previous FIND command (find one or more occurrences of a specific character string) or the FIND part of the most recent CHANGE command (Browse, Edit, and View only).
RIGHT	Scrolls right.
SAREA	Displays the Status Area pop-up window.

Table 3. ISPF System Commands (continued)

Command	Description
SCRNAME *screen name* [PERM|ON|OFF]	Causes the logical screen in which the command is entered to be given the *screen name* specified. The name can be any set of 2 to 8 characters that conform to member naming rules, except NEXT, PREV, LIST, ON, and OFF. PERM is an optional parameter to indicate that ISPF does not allow the SCRNAME parameter on a SELECT statement, or the setting of the modifiable system variable ZSCRNAME, to override the value being assigned by this SCRNAME command. The PERM setting lasts for the duration of the logical screen. After you end the logical screen, the setting is no longer active. SCRNAME ON causes the name that you specify for the screen to be displayed in the *panelid* area of the screen, if panelid is OFF. SCRNAME OFF removes the screen name from visible display.
SETTINGS	Displays the ISPF Settings panel.
SPLIT [NEW]	Causes the screen to be divided into two logical screens separated by a horizontal line or changes the location of the horizontal line. If you have de-selected the *Always show split line* option in Settings, there is no split line in 3270 mode. In GUI mode, each screen is shown as a separate physical window. See "Splitting the Screen Horizontally or Vertically" on page 60 for more information.
SPLITV	On 3290 terminals, causes the screen to be separated into two vertical logical screens. The SPLITV function is not active if the actual screen data display is more than 80 characters wide.
START	Starts a dialog in a new logical screen. If a logical screen does not exist, it will be created. You can use the START command to do the following: • Issue a command from the ISPF command table; for example, START KEYLIST • Issue a command with parameters (in single quotes; for example, `START 'ISRROUTE BRI'` • Start a dialog; for example, `START PANEL(ISRUTIL)` **Notes:** 1. If you are running in GUI mode, issuing the START command creates a new window. 2. If you are running on the host and invoke START from a pull-down choice, the screen will be split where your cursor is located within the pull-down. 3. This function does not change the limitation number of logical screens. If ISPF already has the maximum number of screens when the START command is issued, the screen is re-split; that is, the split line might move on the host, or a window is reused if you are running in GUI mode.

Table 3. ISPF System Commands (continued)

Command	Description
SWAP [LIST | PREV | NEXT | *screen name* | *n*]	When no parameters are given (in 3270 mode), moves the cursor to where it was previously positioned on the other logical screen of a split-screen pair. When operating in split-screen mode, pressing the SWAP key (F9) causes ISPF to ignore any entry on the command line. When no parameters are given (in GUI mode) and there are more than two screens present, defaults to SWAP LIST. Entering SWAP LIST displays the ISPF task list. The task list displays the following information about all of the active logical screens: • Screen ID (ZSCREEN) • Screen name • Panel ID • Application ID • Session type (GUI or 3270) You can select from this list the screen you want to display or start a new screen or application. The screen you select replaces the screen from which you issued the command. Using a listed parameter changes the focus (in GUI mode) or display (in 3270 mode) to the PREVious, NEXT, or specified logical screen. PREV changes the focus or display to the next lower screen number until reaching 1, then wraps back to 32 or the last number used. NEXT displays the next highest screen number until the last number used is reached (ISPF maximum 32, your installation might vary), then will wrap back to number 1. SWAP *screen name* changes the display or focus to the screen called *screen name*, if it is active. See the SCRNAME command for more information about screen names. SWAP *n*, where *n* is a number, changes the display or focus to the specified screen number, if it is active.
SWITCH (3270 | GUI)	Switches the mode of screen display between GUI and 3270.
TOP	Alias for the UP MAX command. Scrolls to the top of the data.

ISPF System Commands

Table 3. ISPF System Commands (continued)

Command	Description
TSO	Allows the user to enter a TSO command, CLIST, or REXX command procedure. Do not enter the following commands after the TSO command: • LOGON, LOGOFF • ISPF, PDF, ISPSTART, and SPF • TEST • Commands that are restricted by TSO or PCF You can enter a CLIST or REXX name after the TSO command, but the following restrictions apply: • The CLIST or REXX command procedure cannot invoke the restricted commands shown in the preceding list. • Restrictions that apply to CLIST attention exits are described in *ISPF Dialog Developer's Guide and Reference* • TERMIN command procedure statements cause unpredictable results.
TSOCMD	Displays the ISPF Command Shell panel.
TSOGUI [ON | OFF]	ON specifies that the ISPF/TSO window is resumed and all TSO line mode output and input is directed to the ISPF/TSO window. OFF specifies that the ISPF/TSO window is suspended and all fullscreen and line mode data appears in the 3270 window until TSOGUI ON command is issued.
TUTOR [panelid]	Calls the ISPTUTOR program to display specified tutorial panels. To display a particular tutorial panel, enter the TUTOR command along with the panel identifier of the desired tutorial panel as a parameter. If you issue the TUTOR command without a parameter, the general tutorial help panel (ISP00000) is displayed.
UP	Scrolls toward the top of the data.
WINDOW	Moves a pop-up that is currently displayed. If more than one pop-up is displayed on your logical screen, only the active (or most recent) pop-up will move. A pop-up can only be moved within the logical screen from which it originated. **Note:** The WINDOW command is disabled if you are running in GUI mode.
WS	Modelessly starts the command you specify, providing a seamless interface between host and workstation applications. For example, when running in GUI mode, you can enter the following command to invoke a workstation editor on file *test.fil*: `WS e.exe test.fil`

Table 3. ISPF System Commands (continued)

Command	Description
WSCON	Displays the *Initiate Workstation Connection* panel. Using this panel, you can start a workstation connection. There are some restrictions to consider if you choose to start a workstation connection with this command: • At the completion of ISPF command processing, or program, clist, or REXX exec execution, ISPF does not reprocess the panel from which the command was invoked. The panel is simply relaunched to the screen without processing, for example, the panel's)INIT section. So, if certain constructs are defined within the panel sections based on the ZGUI variable (non-blank indicating connection in GUI mode), these constructs are not defined properly until after the Enter key is pressed following the WSCON command. • When connecting to the workstation in GUI mode, the first panel displayed in GUI mode might not contain group boxes and images that are defined on the panel. After you press the Enter key, causing the panel to be reprocessed, these constructs are visible in GUI mode.

Table 3. ISPF System Commands (continued)

Command	Description
WSDISCON	Disconnects your workstation connection. There are some restrictions to consider if you choose to disconnect from a workstation with this command: • At the completion of ISPF command processing, or program, clist, or REXX exec execution, ISPF does not reprocess the panel from which the command was invoked. The panel is simply relaunched to the screen without processing, for example, the panel's)INIT section. So, if certain constructs are defined within the panel sections based on the ZGUI variable (non-blank indicating connection in GUI mode), these constructs are not defined properly until after the Enter key is pressed following the WSDISCON command. • When disconnecting from GUI mode, the name of any group boxes defined on the panel from which you issued WSDISCON, will display on the screen in 3270 mode. After you press the Enter key, causing the panel to be reprocessed, these names disappear and any panel text under the names reappears. • If in GUI mode and in split screen mode when you invoke the WSDISCON command, you are disconnected from the workstation and the screen from which you issued WSDISCON is displayed in the full 3270 emulator session without a split line, regardless of any ***Always show split line*** settings. The other ISPF sessions are hidden and available for display after the swap command is entered. Additional split requests cause the split line to be redisplayed, provided the ***Always show split line*** setting is selected. • If popup windows are displayed in GUI mode when the WSDISCON command is issued, those popup windows are suspended on the 3270 session and the panels display as fullscreen panels. If new addpops are then invoked, these new panels display as popups. • You **cannot** disconnect using the WSDISCON command when running in **batch** GUI mode. • If the CODEPAGE and CHARSET parameters were specified (on the WSCON service, or on the Initiate Workstation Connection panel, or on the ISPSTART GUI statement) when your workstation connection was made, these values might no longer be the host codepage and character set in 3270 mode following a WSDISCON command. The values returned from the terminal query are restored as the active codepage and character set. If your emulator does not support codepages, the CODEPAGE and CHARSET parameter values originally specified on your ISPSTART statement are used. If these ISPSTART parameters were not specified, ENGLISH is the default specification.
ZKEYS	Displays a panel that lets you view and change the current function key variables. This command is equivalent to selecting the Global PF Key settings choice from the Function keys pull-down on the ISPF Settings panel.

Using the RETRIEVE, RETF, and RETP Commands

This section describes how to use the RETRIEVE, RETF, and RETP commands.

RETRIEVE Command

The RETRIEVE command causes the most recently entered command to be displayed on the command line. If the command recalled by RETRIEVE is longer than the current primary input field, ISPF truncates the command to the size of the primary input field for display purposes. Only the data displayed in the primary input field is processed and stored in the command retrieval stack when you press Enter or a function key. However, the original command retains its full length in the retrieval stack.

If the current panel has no input fields, then the size of the primary input field is zero and the retrieved command is not displayed. Normal stack processing occurs, however, and the internal pointer is incremented to the next saved command. This can result in an unexpected command being recalled when RETRIEVE is issued on a subsequent panel that has input fields.

If you issue the RETRIEVE command when the stack is empty, ISPF presents you with a blank command line with the cursor in the first position. If the stack is not empty, ISPF places the cursor immediately following the retrieved command.

If you are in the process of recalling a string of commands by issuing successive RETRIEVE commands, you can cause ISPF to recycle to the top of the command retrieval stack by pressing Enter when the primary input field (normally the command line) is blank.

When you are operating in split-screen mode, one stack retains commands for all logical screens.

There are five cases for which ISPF does not retain an entered command for retrieval:

- Commands entered using attention fields, such as light pen-selectable fields or cursor-select fields.
- Commands entered through the use of function keys. This includes any portion of a compound command that results from pressing a function key. For example, if you key PAGE into the primary input field and then press the function key set to the DOWN command, only the PAGE portion of the DOWN PAGE command is retained as a single element in the retrieval stack. The entire character string entered from the primary input field in conjunction with a function key is always retained, whereas any portion of the command resulting from the function key value is not retained.
- The RETRIEVE command, if entered as a single command. If RETRIEVE is one of the commands of a chain being processed by ISPF, the entire chain is placed on the retrieval stack. However, processing of the command chain ends when ISPF interprets the RETRIEVE command and displays the next command in the stack. Any commands following RETRIEVE in the chain are not processed.

  RETRIEVE can be part of a stack element as a parameter of another command. For example, you might enter FIND RETRIEVE as a command.
- Commands entered on the COMMAND option of the DISPLAY service.
- Jump function (extended return) commands entered from a nondisplay field.

You can issue any retrieved command, as is, while it is being displayed, or you can edit the command line and then issue the modified version.

Command retrieval works on a last-in, first-out basis. For example, assume that the last three commands you have issued are PRINT, DOWN, and RIGHT, in that order. Now suppose that you want to again issue the PRINT command. Assuming that F12 is set to RETRIEVE, the sequence of operations is:

1. Press F12. RIGHT displays on the command line.
2. Press F12 again. DOWN displays on the command line.
3. Press F12 a third time. PRINT displays on the command line.
4. Press Enter.

You can also use the RETRIEVE command to check and correct errors made in keying commands. For example, suppose that you mistakenly enter PFSHOW TAYLOR. When ISPF advises you that TAYLOR is not a valid parameter, you would:

1. Press F12. PFSHOW TAYLOR displays on the command line.
2. Type over the Y with an I.
3. Press Enter.

Each ISPF session supports only one command retrieval stack, to be shared by all logical screens. The number of commands that ISPF saves for retrieval depends on:

- The size of the stack area allocated for this purpose by the installation

  Note: Refer to *ISPF Planning and Customizing* for the appropriate material on changing the size of the stack area allocated for RETRIEVE command processing.
- The lengths of the individual command lines that are saved.

As a command is entered, it goes to the top of the stack, pushing all other commands down. If there is not enough room at the bottom of the stack to hold the entire bottom command, it is dropped from the stack.

Duplicate commands are allowed in the stack, except when the command being entered is a duplicate of the command at the top of the stack. All command lines (except the RETRIEVE command, as explained previously) are placed in the stack as entered, regardless of validity. Actually, these commands can be any character string, up to 255 bytes each, entered from the screen's primary input field (not necessarily the ZCMD field).

Jump function commands are stored in the stack unless they are entered from a nondisplay field, regardless of whether the field is the primary input field or not.

If the RETRIEVE command is repeatedly entered until the bottom command in the stack displays, issuing the RETRIEVE command once more causes the command at the top of the stack to be displayed again. To force a return to the top of the stack, clear the command field and press Enter. Then, the next RETRIEVE command causes the command line to be set to the command at the top of the stack.

RETF Command

The forward retrieve (RETF) command recalls commands from the command retrieval stack from the oldest command in the stack towards the most recent commands in the stack. This is useful when you RETRIEVE too many times in an attempt to retrieve a specific command. RETF enables you to return to the desired command without having to cycle through the entire retrieval stack.

RETP Command

The retrieve pop-up (RETP) command causes a pop-up panel to be displayed with the last 20 commands in the command retrieval stack listed. You can select the

command you want to retrieve by number. The selected command is retrieved to the command line. When using the RETP command, the following considerations apply:

1. If a command in the command retrieval stack is too long to fit in the retrieve pop-up, the last visible character of the command is changed to a > to show that some characters are not displayed. However, the entire command is retrieved to the command line when it is selected to be retrieved.
2. The default for the minimum number of characters is one, so any command entered is saved on the retrieval stack. The user has the option of setting the value from 1–99 for the minimum number of characters to save. Therefore, if you select three characters for the minimum number to be saved in the retrieval stack and a one- or two-character command is entered, it is not added to the retrieval stack. This prevents short commands that can be easily retyped from taking up space in the retrieval stack. Changing the minimum number of characters to save in the retrieval stack does not affect commands already in the retrieval stack. This setting is saved in the variable ZRETMINL, which is saved in the user's ISPF system profile table ISPSPROF and across ISPF invocations.
3. The default for the cursor position when a command is retrieved is at the end of the command. The cursor position setting is saved in the variable ZRETPOSC, which is saved in the user's ISPF system profile table ISPSPROF and across ISPF invocations.

Using the RETURN Command

The RETURN command causes the immediate return to a primary option menu or to the display from which you entered a nested dialog. When a RETURN command is entered, the DM component takes the following action:

1. It simulates the END command on the panel that is currently displayed; that is, the DISPLAY or TBDISPL service returns a code of 8.
2. For subsequent requests, made through the DISPLAY or TBDISPL service, for display of a different panel, the panel is not displayed, and a return code of 8 is issued by the service.
3. However, when two consecutive display requests name the same panel, normal operation of the DISPLAY and TBDISPL services is restored and processing proceeds as though RETURN had not been entered. The DM component decides whether to proceed. Generally, because RETURN signals the application user's desire to end the current processing, a developer can limit processing after the RETURN is received to clean up and do final processing before returning control to the dialog element from which the function was started.
4. If two consecutive requests do not specify the same panel, processing continues in the mode described in item 2 until control is returned to a primary option menu or a nested dialog completes. Then, normal operation of the DISPLAY and TBDISPL services is restored.

It might be necessary to suspend processing of a panel temporarily so that other panels can be displayed. Issue a CONTROL DISPLAY SAVE request to save the contents and control information of the panel whose processing is to be suspended. Before resuming the processing of this panel, issue CONTROL DISPLAY RESTORE to reinstate the contents and control information for the panel. If non-ISPF screens have been displayed, issue CONTROL DISPLAY REFRESH to clear the screen.

This mode of operation continues until either a primary option menu is encountered or a nested dialog completes. If a primary option menu is

encountered, it is displayed. If a nested dialog completes, the panel from which it was invoked is redisplayed. This panel is exactly as you last saw it, except that the command field is blank. In either case, this completes the action of the RETURN command.

Note: A nested dialog is one invoked from any panel by a SELECT action command. The HELP and KEYS commands invoke nested dialogs. In addition, the TSO system commands invoke nested dialogs when they are used to execute a CLIST procedure that displays panels through ISPF services.

If a dialog function needs to distinguish between END and RETURN, it can do so in one of the following ways:

- If the panel was defined using the panel definition statements, upon return from the DISPLAY or TBDISPL service, with a return code of 8, the function can examine variable ZVERB in the shared pool. It contains either END or RETURN.
- If the panel was defined using the DTL, upon return from the DISPLAY or TBDISPL service, with a return code of 8, the function can examine variable ZVERB in the shared pool. It contains either EXIT or CANCEL.
- Upon return from the SELECT service when the PANEL keyword was specified, the dialog function can examine the return code from SELECT. Return code 0 indicates that the END command was entered on the selected menu panel. Return code 4 indicates that the RETURN command was entered on the selected menu panel or on some lower-level menu.

Using the Jump Function

The jump function allows you to go directly to any valid option from the primary option menu currently in effect. Refer to *ISPF Dialog Developer's Guide and Reference* for information on coding primary option menus. To use the jump function, enter the option on the command line or in the command field of any panel, preceded by an equal sign and followed by a blank. For example:

```
Command ===> =3.1
```

takes you directly to the first suboption of option 3 on the primary option menu in effect.

The action is as follows:

- If not entered on a primary option menu, the jump function causes repeated END commands to be simulated until a primary option menu is encountered. What follows the equal sign is then used on the primary option menu, and pressing of the Enter key is simulated. The primary option menu is not displayed.
- If entered on a primary option menu, the jump function equal sign is ignored and the specified option is selected.

Unlike the RETURN command, the jump function is not affected by nested dialogs. For example, from the ISPF Edit option, you enter a HELP command to enter the tutorial. Then from the tutorial, you enter =1. This causes the tutorial to end, Edit to end, and primary option 1 to be started.

For convenience, you can enter a jump function in two other places:

- Any field that is preceded by an arrow. The arrow must consist of at least two equal signs followed by a greater-than sign (==>). Also, the arrow must immediately precede the input attribute byte.

- Any field preceded by leader dots (that is, ... or . .). ISPF looks at the three characters preceding the field; they must be either three consecutive dots or two dots separated by a blank.

The command field is the only field that can be initialized to `=n` by the dialog and have the jump function recognize it. Modifying the ZCMD field in the)PROC or)INIT section can affect jump function operation.

If ISPF encounters an error during jump function processing, the processing stops with the jump function in error displayed on the command line, unless that function was entered from a nondisplay field.

Because a jump request generally signals a user's desire to end the current processing, the dialog developer must limit processing to cleaning up and completing processing before returning control to the selection in the jump request. Otherwise, the dialog developer can cancel the jump request/return mode by providing two consecutive displays with the same panel name.

For compatibility with the SPF licensed program, the jump function can be entered in conjunction with the RETURN command or RETURN function key. For example, you type `=2` and then press the RETURN function key rather than pressing Enter. The action is just the same as if you had typed `=2` and pressed Enter.

Using the Scrolling Commands

You can use the scrolling commands if the dialog function invokes the DISPLAY service for panels with scrollable areas or scrollable dynamic areas, the table display service (TBDISPL), or the interfaces to the PDF component VIEW, BROWSE, and EDIT services. During processing of the tutorial, ISPF interprets these commands as follows:

UP (F7/19)
: Same as the UP command

DOWN (F8/20)
: Same as the SKIP command

LEFT (F10/22)
: Same as the BACK command

RIGHT (F11/23)
: Same as the Enter key (display the next page).

When scrollable data is displayed, scrolling enables you to move the screen *window* up, down, left, or right across the information. Only up and down scrolling is allowed for table displays and scrollable areas.

When scrolling is allowed, a scroll amount is commonly displayed at the top of the screen (line 2). This amount determines the number of lines, or columns, scrolled with each use of a scroll command. To change the scroll amount, move the cursor to the scroll field and type over the displayed amount. Valid scroll amounts are:

0–9999 Specifies the number of lines (up or down) or columns (left or right) to be scrolled.

Note: If you specify a scroll amount of **0**, no scrolling occurs.

PAGE Specifies scrolling by one page.

For scrolling purposes, a *page* is defined as the amount of information currently visible on the logical screen. Function key definition lines are not a part of the page. In split-screen mode, for example, a Browse display

might have 12 lines by 80 columns of scrollable data. In this case, a scroll amount of PAGE moves the text up or down by 12 lines, or right or left by 80 columns.

DATA For up and down scrolling, specifies scrolling by one line less than a page. For left and right scrolling, it is one column less than a page.

HALF Specifies scrolling by half a page.

MAX Specifies scrolling to the top, bottom, left margin, or right margin, depending upon which scrolling command is used.

CSR Specifies scrolling based on the current position of the cursor. The line or column indicated by the cursor is moved to the top, bottom, left margin, or right margin of the screen, depending upon which scrolling command is used. If the cursor is not in the body of the data or if it is already positioned at the top, bottom, left margin, or right margin, a full-page scroll occurs.

Note: Scroll amount is not used for scrollable areas.

The current scroll amount is saved in the application profile. There are three scroll amount values: one for Browse (ZSCBR), one for Edit and View (ZSCED), and one for member lists (ZSCML). When you type over the scroll amount, the new value remains in effect until you change it again. The value MAX is an exception. Following a MAX scroll, the scroll amount reverts to its previous value.

The scroll amount field is optional. If the input field following the command field in the panel body is exactly four characters long, it is assumed to be the scroll amount field. Otherwise, the system variable ZSCROLLD, which can be set by the dialog, is used to determine the default scroll amount. If there is no scroll amount field and ZSCROLLD has not been set, the default is PAGE.

When you enter a scroll request, the variables ZSCROLLA and ZSCROLLN are set. ZSCROLLA contains the value of the scroll amount field, such as MAX or CSR. ZSCROLLN contains the number of lines or columns to scroll, computed from the value in the scroll amount field. For example, if a dialog is in split-screen mode, 12 lines are currently visible and you request DOWN HALF, ZSCROLLN contains a 6. The system variable ZVERB contains the scroll direction (DOWN in this case). If the value of ZSCROLLA is MAX, the value of ZSCROLLN is not ignored.

You can also use any valid scroll amount as part of the scroll command. For example, type:

```
Command ===> UP 3
```

and press Enter, or type:

```
Command ===> 3
```

and press the UP function key. Either form results in a temporary, one-time override of the scroll amount.

If ISPF does not recognize the value specified on the command line as a valid scroll amount, such as PAGE, DATA, HALF, MAX, CSR, or a positive integer, the value is interpreted as a command and passed to the function in control.

Using Command Tables to Define Commands

ISPF implements system, user, site, and application commands through the use of command tables.

A system command table (ISPCMDS) is distributed with ISPF in the table input library. An application can provide an application command table by including a table named *xxxx*CMDS in its table input library, where *xxxx* is a 1- to 4-character application ID. You can also add your own user command table to the ISPF Configuration table. This is a permanent place for your set of user defined commands. When IBM updates the ISPF command table, you do not need to re-add your commands. By setting the Before or After option, you can search this command table either before or after the ISP command table. The default option is Before. If the application's table input library is defined with the LIBDEF service, the LIBDEF must be active when the SELECT service call that invokes the application is issued, and the PASSLIB parameter must be specified.

You can define an application command table using either:

- The command table utility described in the **Command Table Utility (Option 3.9)** section of the *ISPF User's Guide Volume II*.
- The Dialog Tag Language (DTL) and ISPF conversion utility. Refer to the *ISPF Dialog Tag Language Guide and Reference* for the tags you must use.

When a user enters a command, the DM component searches the application command table, if any, then the user command table, if any, and then the system command table, ISPCMDS. This is the default search order, which assumes the option Before for the user command table. If you choose the option After for this table, it is searched after ISPCMDS. If it finds the command, action is taken immediately. If it does not find the command in the application or system tables, the command is passed to the dialog, unaltered, in the command field. The dialog must then take appropriate action.

Command Table Format

A command table is an ISPF table in which each row contains the specification for one command. Each column contains a variable for the command. The variables are:

ZCTVERB
: Specifies the name of the command. A command name must be from 2–8 characters long and must begin with an alphabetic character. Note that the terms *command name* and *command verb* are synonymous and are used interchangeably.

ZCTTRUNC
: Specifies the minimum number of characters that you must enter to find a match with the command name. If this number is zero or equal to the length of the name, you must enter the entire name. This number must not be one, or be greater than the length of the name.

ZCTACT
: Specifies the action to be performed when the command specified in ZCTVERB is entered. Can be up to 240 characters.

ZCTDESC
: Contains a brief description of the purpose of the command. This variable is optional. It is not used by the DM component in processing the command, but it is displayed by the command table utility. The description is limited to 80 characters.

The dialog manager treats ZCTVERB, ZCTTRUNC, ZCTACT, and ZCTDESC as defined function variables. They are not accessible to dialogs.

The valid actions that can be performed (ZCTACT) are:

SELECT
Followed by selection keywords causes the selected dialog (command, program, or menu) to be given control immediately.

ALIAS
Followed by another command and any parameters allows specification of command aliases.

PASSTHRU
Causes the command to be passed to the dialog instead of continuing to search the system table.

SETVERB
Causes the command to be passed to the dialog with the command verb stored in ZVERB separately from the parameters. The ISPF system commands distributed with the product that have SETVERB as an action are not always passed through to the dialog. Refer to "Passing Commands to a Dialog Function" on page 116 for further discussion.

NOP Causes the command to be inactive. ISPF displays an inactive command message in this case.

Blank (no action)
Causes the table entry to be ignored. Scanning continues, searching for additional entries with the same name.

A variable name
The name begins with an ampersand and can be one of the actions described in this list. This allows dynamic specification of command action.

Additional action keywords are used to indicate system commands for which special processing is required. These are CURSOR, PRINT, PRINTG, PRINT-HI, PRINTL, PRINTLHI, SPLIT, SPLITV, SWAP, and RETRIEVE. Although these are valid actions, they are intended for use only in the system command table distributed with ISPF and are intended to be used only with the associated command verb. They are not intended for use in application command tables.

Customizing the ISPCMDS Command Table

Use the following steps to customize your ISPCMDS command table:

1. Copy the ISPCMDS into a dataset concatenated prior to the '*.SISPTENU' dataset in the ISPTLIB DD statement.
   - Name the new member using a unique prefix of up to 4 characters, for example: RSMCMDS or MOD1CMDS.
   - Allocate the copied-to dataset to the ISPTABL DD card. You can use the following CLIST to do a LIBDEF against ISPTABL if you have a ISPTABL DD allocated:

     ```
     PROC 0
     ISPEXEC LIBDEF ISPTABL DATASET ID(the_dataset_name)
     WRITE &LASTCC
     END
     ```

     If your logon procedure does not allocate an ISPTABL DD card:

```
PROC 0
ALLOC F(ISPTABL) DA(the_dataset_name)
END
```

2. Using option 3.9, customize this member with your new commands.
   - Option 3.9 will search the ISPTLIB DD for this member.
   - Option 3.9 will save (UPDATE) this customized member to the output data set pointed to by ISPTABL.
3. After you have customized this member, you can use option 3.1 or 3.4 member list to rename this member to ISPCMDS.
4. Exit ISPF.
   - This will nullify the LIBDEF on ISPTABL.
5. Re-invoke ISPF.
   - When ISPF searches the ISPTLIB DD concatenation, your customized ISPCMDS will be found first.

SELECT Action Commands Temporarily Invoke a New Dialog

A SELECT action command can be specified in a command table. The action is coded exactly the same as for the SELECT service. All SELECT keywords are valid, including NEWAPPL.

The selected dialog is started immediately when a SELECT action command is entered on the command line of any panel. This temporarily suspends the current dialog. When the selected dialog completes, the screen is refreshed and the suspended dialog resumes.

Examples of SELECT action commands:

ZCTVERB	ZCTTRUNC	ZCTACT
UPDATE	0	SELECT PGM(PQRUPDT) PARM(&ZPARM)
PREPARE	4	SELECT CMD(XPREP &ZPARM) NEWPOOL
MENU	4	SELECT PANEL(&ZPARM)

In the example, the ZCTTRUNC variable indicates that the UPDATE and MENU command names cannot be truncated. PREPARE, however, can be truncated to PREPAR, PREPA, or PREP. The functions and keywords in the ZCTACT field indicate the actions that the commands perform.

The ZPARM variable that appears in the SELECT keywords indicates that command parameters are to be substituted at that point. For example, if the following commands were entered:

```
===> UPDATE BLDG DEPT NAME
===> PREPA LOG LISTING
===> MENU PQRMENU1
```

the following SELECT actions would result:

```
SELECT  PGM(PQRUPDT) PARM(BLDG DEPT NAME)
SELECT  CMD(XPREP LOG LISTING) NEWPOOL
SELECT  PANEL(PQRMENU1)
```

ZPARM, a dummy variable, is used only to substitute user-entered parameters into SELECT action commands. It is not stored in a variable pool and is not accessible to dialogs.

Note: Use of SELECT action commands can cause recursive entry into dialog functions, which the DM component allows. The dialog developer should either design functions for recursive use or display a message if a user attempts to reenter a nonrecursive function.

The ISPF DISPLAY and TBDISPL services can be used recursively. The current display environment is automatically saved whenever a SELECT action command is entered and is restored upon completion of the command.

Assigning Command Aliases

A command alias is an alternate way of expressing a command. For example, you might assign to the command UP MAX an alias of TOP to make it easier to remember and to issue. In the case of a command that includes lengthy parameters, using an alias can be a much more efficient way of entering the command. Also, using aliases can be helpful for writing dialogs in languages for which single words can meaningfully replace multiword command-parameter expressions. Normally, alias entries are used in an application command table to refer to system commands, which might or might not include parameter fields. Issuing the command or its alias causes the same result.

An alias must precede, in the command table, any reference to the command to which it refers. You can establish an alias by setting values in two command table variables. Set:

- ZCTVERB to the value you wish to use as the alias for an existing command
- ZCTACT to the keyword ALIAS followed by the command, including any parameters, for which you are establishing the alias. Thus, the value of the ZCTACT variable can be either a single-word command, such as HELP, or it can be a multipart command, such as UP MAX.

You can set the value of ZCTTRUNC in the command table to the minimum number of characters of the alias name that must be entered. For example, for the alias FORWARD, if you set ZCTTRUNC to a value of 3, issuing the first three characters (or more) has the same effect as issuing FORWARD. If you assign a value of 0 in the ZCTTRUNC field, the complete alias name must be issued.

The maximum length of the value you can specify in ZCTACT, including the keyword ALIAS, followed by a blank, and the command verb plus any parameters, is 240 characters. This leaves a maximum of 234 characters for the command, at least one blank following the command, and any parameters. ISPF interprets up to the first 8 characters in the command name. ISPF issues an error message for command names that do not contain from 2–8 characters.

Any parameters included in the ZCTACT command table field take precedence over any parameters included with that command's alias issued from a command line. Thus, if you issue a command alias that includes parameters, ISPF:

- Recognizes the command alias verb
- Ignores the parameters you specified with the alias
- Substitutes the parameters included with the command verb in ZCTACT.

If the command verb in ZCTACT does not include parameters, ISPF accepts parameters specified with the command's alias from a command line. This can be useful when a command's parameters do not fit into the 240 character spaces available in ZCTACT.

You can create a chain of command-parameter aliases in a command table as long as the result is a valid executable action. The last command verb and parameter

values that ISPF encounters in the alias chain within the command table are the ones that are executed. The command verb and the parameter values do not necessarily come from the same table entry. For example:

ZCTVERB	ZCTTRUNC	ZCTACT
EASYKEY	0	ALIAS CMD PARM1 PARM2
CMD	0	ALIAS CMD1 PARM3
CMD1	0	ALIAS CMD2

In this example, if you entered EASYKEY from a command line, the command that would ultimately be executed would be CMD2 PARM3.

Some other examples of defining alias values are:

ZCTVERB	ZCTTRUNC	ZCTACT
QUIT	0	ALIAS END
FORWARD	3	ALIAS DOWN
*TOP	0	ALIAS BACKWARD MAX
*BACKWARD	0	ALIAS UP
*ENDFILE	4	ALIAS LOW
*LOW	0	ALIAS DOWN MAX

*These four entries represent two-level chaining.

This example defines QUIT as an alias of END, FORWARD as an alias of DOWN, and so on. For example, if you enter QUIT, the system responds as though you had entered END.

Looking at the two-level chaining examples, if you enter TOP, ISPF responds as though you had entered UP MAX. This is because, at the second level when BACKWARD is replaced with UP, there is no second-level parameter to replace MAX. In the case of ENDFILE, ISPF responds as though you had entered DOWN MAX. ISPF replaces the verb LOW with DOWN and the blank parameter value with MAX.

Note: Command aliases shipped with ISPF in table ISPCMDS include TOP (UP MAX), BOTTOM (DOWN MAX), BACKWARD (UP), and FORWARD (DOWN).

Overriding System Commands

An application can override any system command simply by including the same command name in the application command table. For example:

ZCTVERB	ZCTTRUNC	ZCTACT
HELP	0	PASSTHRU
TSO	0	NOP

In this example, the dialog has overridden both the HELP and TSO commands. During ISPF processing, if you enter HELP, the command is passed to the dialog function in control, which determines the action to be taken. The action specified for the TSO command is NOP, which disables the TSO command. ISPF displays an inactive command message when a NOP action command has been processed.

Passing Commands to a Dialog Function

As previously noted, any command that is not found in the application or system command table is passed, unaltered in the command field, to the dialog. This occurs regardless of whether the command was typed in the command field or entered by use of a function key or the attention field.

You can force a command to be passed to the dialog, even if the command exists in the command table, by typing a greater-than symbol (>) in front of the command.

Any command in the command table that has an action of PASSTHRU is processed as though the command were not found in the table. It is passed in the command field to the dialog.

Commands can also be passed to the dialog using the SETVERB action. This action causes the dialog manager to separate the name from the command parameters, if any. The command is stored in variable ZVERB, which is in the shared pool. The left-justified command parameters are passed in the command field to the dialog. For example:

ZCTVERB	ZCTTRUNC	ZCTACT
QUERY	0	SETVERB

The verb QUERY is stored in variable ZVERB and the character string, such as `DEPT 877` in the following examples, is passed in the command field.

The following actions produce the same results:

- Typing `QUERY DEPT 877` in the command field and pressing Enter.
- Typing `DEPT 877` in the command field and pressing a function key that has been equated to the character string `QUERY`.
- Pressing a function key that has been equated to the character string `QUERY DEPT 877`.
- Using the light pen or cursor-select key to select an attention field that contains the character string `QUERY DEPT 877`.

The following system commands, distributed with the DM component, are defined as SETVERB action commands:

END	UP
RETURN	DOWN
RFIND	LEFT
RCHANGE	RIGHT

The ZVERB variable can be used to distinguish between END and RETURN. The effect of END and RETURN on the DISPLAY service is the same because RETURN is used to simulate repeated END commands, until a primary option menu is reached.

RFIND and RCHANGE are used only by ISPF View, Browse, and Edit. Thus, these commands are not passed back to a user dialog in ZVERB.

The commands UP, DOWN, LEFT, and RIGHT are only active when a scrollable panel is displayed. Use of these commands from a nonscrollable panel results in a command is not active message.

Specifying Command Actions Dynamically

You can specify a command action dynamically (as part of function processing) by the use of a dialog variable. A variable action can be used to *share* commands, such as UP, DOWN, LEFT, and RIGHT, with the DM component. It can also be used to enable or disable commands during certain points in the dialog. Suppose, for example, an application command table includes the following entries:

ZCTVERB	ZCTTRUNC	ZCTACT
UP	0	&SCRVERT
DOWN	0	&SCRVERT

You can use the variable SCRVERT to dynamically control the action of the UP and DOWN vertical scroll commands as follows:

- If SCRVERT is set to NOP, the commands are not available.
- If SCRVERT is set to PASSTHRU, the commands are passed to the dialog.
- If SCRVERT is set to blank, command scanning continues. In this case, the system definitions for UP and DOWN in the system command table take effect.
- If SCRVERT is set to an action that is not valid, the commands are not available, as in NOP.

For this particular example, setting SCRVERT to SETVERB would have the same effect as setting it to blank, because UP and DOWN are defined in the system command table as SETVERB action commands.

If the dialog overrides or shares the use of the scroll commands, it becomes that dialog's responsibility to ensure that the commands have been redefined with an action of blank, or with SETVERB. This must be done before starting any ISPF function that requires View, Browse, Edit, and Table Display. The same rule applies to the RFIND command used by Browse and Edit and the RCHANGE command used by Edit.

Using a Function Key

Under ISPF, function keys are not automatically assigned to special functions. You equate each function key to a character string. When you press a function key, it simulates command entry. The processing is the same as if you had typed the character string in the command field and pressed the Enter key.

Note: On a 3270 display, the horizontal divider line that separates the logical screens is not considered part of either logical screen. If the cursor is placed on this horizontal divider line and a function key is pressed, the result is the same as if the ENTER key was pressed and the cursor is positioned on the active logical screen's command line.

A dialog function cannot distinguish the difference between a command entered by a function key and a command entered by typing in the command field. If the character string with which the function key is equated is longer than the screen's command field, the string is truncated without warning.

Using a Function Key

If you type information on the command line and then press a function key, the function key definition, followed by a blank, is concatenated ahead of the contents of the command field. For example, suppose F7 is equated to the character string `UP`. If you type `4` in the command field and then press F7, the results are exactly the same as if you had typed `UP 4` in the command field and pressed the Enter key.

ISPF does not require function keys for its operation. Commands can be entered in the command field of any display, including View, Browse, Edit, and Table Display. However, for ease of use, function keys are strongly recommended.

The default function key assignments distributed with ISPF for the 3x4 key pad on the right side of the keyboard are shown in Table 4. These are function keys 1–12 on a 12-key terminal or keys 13–24 on a 24-key terminal.

Table 4. Function Key Arrangement

F1	HELP
F2	SPLIT
F3	END
F4	RETURN
F5	RFIND
F6	RCHANGE
F7	UP
F8	DOWN
F9	SWAP
F10	LEFT
F11	RIGHT
F12	RETRIEVE

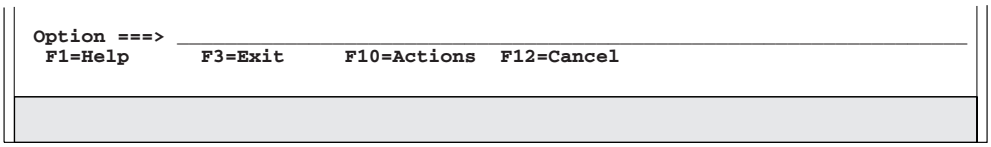
Function keys can be displayed at the bottom of a panel. Using the FKA or PFSHOW command, you can display either the long or short form of the keys, or remove the keys from the panel. See “ISPF System Commands” on page 88 for a complete description of how to display or remove the function keys.

For panels defined without the)PANEL section, the long and short form of the function key area is the same. If you use a)PANEL section, you can use the KEYLIST command or the **Keylist settings...** choice from the Function keys pull-down on the ISPF Settings panel to determine which keys appear in each form. See the **Keylist Settings** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for more information.

Long Displays the keys that appear in the short form along with all other keys you indicated should appear for the long form. The long form is the default. An example of the long form follows:

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

Short Displays the keys that appear in the short form. An example of the short form follows:



No Removes the function key area, making the space available for the application. The keys are still active but are not displayed.

Defining Function Key Values

You can define function key values three ways:

- Use the KEYS command to display the Keylist Utility panel or the PF Key Definitions and Labels panel, then change the function keys for the panel you are on.
- Use the ZKEYS command or select the **Non—Keylist PF Key settings** choice from the Function keys pull-down on the ISPF Settings panel. Use this method to define the function keys when the)PANEL statement has been coded on the panel. All DTL-generated panels have a)PANEL statement. See the **Working with Function Keys and Keylists (The Function Keys Action Bar Choice)** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for more information.
- Use the KEYLIST command or select the **Keylist settings...** choice from the Function keys pull-down on the ISPF Settings panel. Use this method to define the function keys when the application panels are defined with the DTL. See the **Keylist Settings** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for more information.

Changing the Format of the Function Key Area

The FKA and PFSHOW commands let you change the visual display of the function keys on a panel. You can display the long form, short form, or remove the keys completely. You can also use the PFSHOW command with the TAILOR parameter or the **Tailor function key display...** choice from the Function keys pull-down on the ISPF Settings panel to display the Tailor Function Key Definition Display panel shown in Figure 37 on page 120.

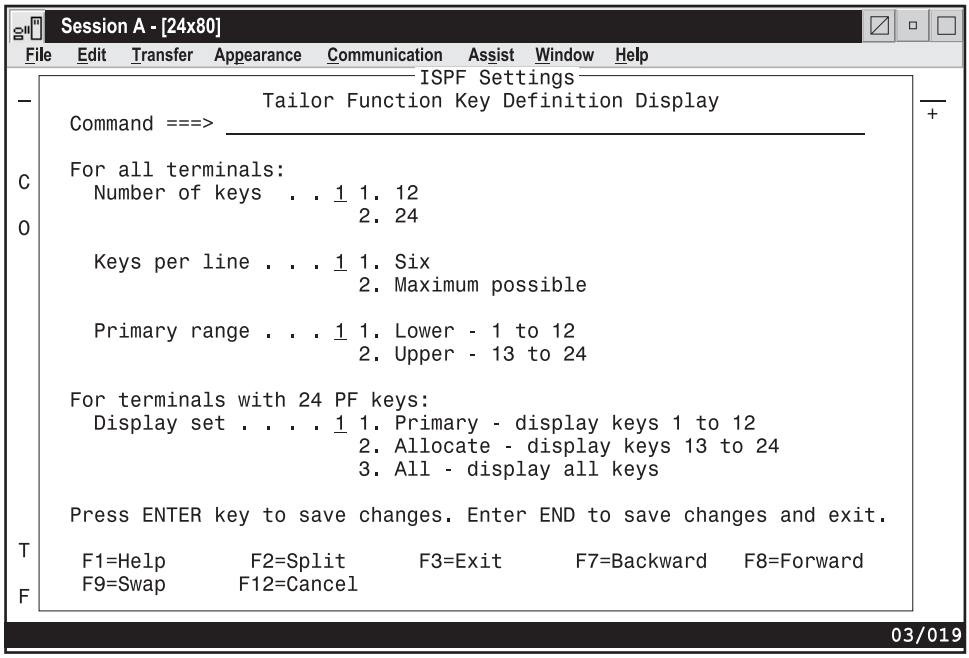


Figure 37. Tailor Function Key Definition Display Panel (ISPOPFA)

This panel lets you select:

- The number of function keys available for display.

  See the **Tailor Function Key Definition Display** section of the ***Settings (Option 0)*** chapter of the *ISPF User's Guide Volume II* for more information. for a discussion of the rules governing the number of keys available for display.
- The number of keys per line to display in each function key definition line. System variable ZPFFMT holds the value selected.

Choice	Description	ZPFFMT value
Six	Always displays six keys per line	SIX
Maximum possible	Displays as many keys as will fit on each line.	MAX

Note: The Maximum possible option is forced when you select the Panel display CUA mode option on the ISPF Settings panel.

- The set of function keys that are to be the primary and alternate keys. System variable ZPRIKEYS holds the value selected.

Choice	Description	ZPRIKEYS value
Lower - 1 to 12	Primary keys are 1–12	LOW
Upper - 13 to 24	Primary keys are 13–24.	UPP

The default value is Lower - 1 to 12.

- The set of function keys on terminals with 24 function keys for which definitions are to be displayed. System variable ZPFSET holds the value selected.

Choice	Description	ZPFSET value
Primary - display keys 1 to 12	Primary set (1–12)	PRI

Choice	Description	ZPFSET value
Alternate - display keys 13 to 24	Alternate set (13–24)	ALT
All - display all keys	All keys (1–24).	ALL

ISPF ignores these values for terminals with only 12 function keys.

Variables ZPFFMT, ZPRIKEYS, and ZPFSET are stored in the application profile pool. Dialogs can set these values directly by using the VPUT statement in a panel definition, or by using the VPUT service in a dialog function.

Dialog developers can effect the way the PFSHOW command behaves by using the ZPFCTL system variable. ZPFCTL is also stored in the application profile pool. Its possible values are:

USER The user can control the display of function key definition lines by using the PFSHOW command. This is the default value.

ON ISPF unconditionally displays function key definitions on all panels. Issuing PFSHOW OFF, FKA OFF, or toggling to the *no display* setting causes ISPF to issue an error message.

OFF ISPF does not display function key definition lines. If PFSHOW ON, PFSHOW TAILOR, FKA ON, or toggling to the long or short form setting of either command is issued, ISPF displays an error message.

Applications can set the ZPFCTL variable value to either USER, ON, or OFF by using the VPUT service or by using a VPUT statement with the PROFILE keyword.

Note: The ZPFCTL variable is ignored if the PFSHOW/FKA command is invoked from a panel containing a)PANEL statement or if the panel was created using DTL.

Similarly, keylists can be controlled to some degree by the application.

The ZKLUSE can be set to Y or N. If KEYLIST is ON, the value in ZKLUSE in the application profile is Y. If KEYLIST is OFF, it is N. If an application VPUTs the variable to the application profile, the keylist setting is altered.

The following variables can be used by an application to determine what keylist is being used, and where it comes from.

- ZKLNAME - If KEYLIST is ON and it is a panel with the)PANEL statement, ZKLNAME contains the name of the keylist currently being used. Otherwise it is blank.
- ZKLAPPL - If KEYLIST is ON and it is a panel with the)PANEL statement, ZKLAPPL contains the application ID that the keylist currently being used came from. Otherwise it is blank.
- ZKLTYPE - If KEYLIST is ON and it is a panel with the)PANEL statement, ZKLTYPE contains either P (private) or S (shared), depending on the keylist currently being used. Otherwise it is blank.

Function key definitions appear at the bottom of each logical screen. There can be more than one logical screen, such as when you are using the split-screen function. If the application has not issued an ADDPOP service call, ISPF displays no more than four function key definition lines on one panel. If the application has issued

an ADDPOP service call and the set of keys to be displayed is primary or alternate, ISPF displays no more than two lines. If all of the keys are to be displayed, no more than four lines appear. If all of the keys will not fit on the lines, ISPF wraps the keys and truncates the last keys.

Saving Function Key Definitions

This section applies only if you created your application panels using the ISPF panel definition statements and used the ZKEYS command or selected the **Global PF Key settings** choice from the Function keys pull-down on the ISPF Settings panel. It does not apply for keys defined with the KEYLIST command or through the **Keylist settings...** choice from the Function keys pull-down on the ISPF Settings panel.

Function key definitions are kept in a set of system variables named ZPF01, ZPF02, ... ZPF24. Labels are kept in a set of system variables named ZPFL01, ZPFL02, ... ZPFL24.

When you set the **Primary range** field on the Tailor Function Key Definition Display panel to Upper - 13 to 24, variables ZPF13–ZPF24 and ZPFL13–ZPFL24 contain the *primary* PF key definitions and labels. For 24-key terminals, these definitions correspond to physical keys 13–24. For 12-key terminals, these definitions correspond to physical keys 1–12. Variables ZPF01–ZPF12 contain the *alternate* key definitions, and are meaningful only for terminals with 24 function keys.

When you set the **Primary range** field on the Tailor Function Key Definition Display panel to Lower - 1 to 12, variables ZPF01–ZPF12 and ZPFL01–ZPFL12 contain the *primary* PF key definitions and labels. For 24-key terminals, these definitions correspond to physical keys 1–12. For 12-key terminals, these definitions correspond to physical keys 1–12. Variables ZPF13–ZPF24 contain the *alternate* key definitions and are meaningful only for terminals with 24 function keys.

Current values for all 24 keys (variables ZPF01–ZPF24 and ZPFL01–ZPFL24) are kept in the application profile. Hence, unique function key definitions can be associated with different applications.

An application can provide default function key settings for a new user by providing a default profile. An application can prevent the user from changing the default function key settings by overriding the ZKEYS command. It does this by assigning the command to NOP in the application command table.

Using the Light Pen and Cursor-Select Key

ISPF permits fields on a panel to be detected with a light pen or the cursor-select key. The cursor-select key is a hardware feature on 3179, 3179G, 3180, 3278, 3279, and 3290 terminals. Only the *attention* mode of light pen selection is used.

Panel fields that are detectable by light pen or cursor selection can simulate a command entry, or give you an alternate means of selecting options from a menu. Each field must be defined as an attention field. Use an attribute character that has been defined with the ATTN(ON) keyword. The panel designer must provide the number of blank characters that are required by the terminal hardware before and after the attention attribute character.

Processing of light pen or cursor-selected fields is handled in much the same way as function key processing. The entire contents of the selected field are treated as a

command and processed as though they had been typed into the command field. If the command is found in the tables, it is performed immediately. If the command is not found in the tables, it is inserted into the command field, and the entire command field is passed to the dialog. But unlike function keys, information in the command field is not concatenated with the contents of the attention field. They should not be used on data entry panels, because any information that is typed in an input field, including command fields, is lost when the attention occurs.

Attention fields can be used on a menu to simulate option selection. The panel designer must truncate any unwanted characters resulting from an attention entry into the command field. An example is shown in Figure 38.

```
)ATTR
  $ TYPE(TEXT) ATTN(ON)
)BODY
%------------------------------ SOME MENU ------------------------------
%SELECT OPTION ===>_ZCMD                                               +
%
$   1 - BROWSE   +DISPLAY SOURCE DATA OR LISTINGS
$   2 - QUERY    +FIND OUT INFORMATION ABOUT SOMETHING
.
.
.
.
.
)PROC
 &ZCMD = TRUNC (&ZCMD, ' ')
 &ZSEL = TRANS (TRUNC (&ZCMD, '.')
                 1, 'PGM(ISPBRO)'
                 2, 'PANEL(XYZ)'
                 .
                 .
                 .
```

Figure 38. Use of Light Pen Attribute

In Figure 38 a light pen or cursor-selection of the first option would place the character string 1 – BROWSE in the ZCMD field and simulate the Enter key. In the)PROC section, the contents of the ZCMD field are truncated at the first blank before the ZSEL variable is set, based on a translation of the ZCMD field.

Panels that are shipped with the ISPF product do not contain the ATTN(ON) keyword in the attribute section. If light pen or cursor selection is used, it is the user's responsibility to add the ATTN(ON) keyword to the attribute section of the desired panel. Refer to the *ISPF Dialog Developer's Guide and Reference* for complete descriptions of the various panel sections.

How Program Access (PA) Keys Affect ISPF Operation

The two Program Access (PA) keys are defined as follows. These definitions cannot be changed.

ATTENTION (PA1)
: Normally, you should not use this key while you are in ISPF full-screen mode. The text following discusses exceptions.

RESHOW (PA2)
: Redisplays the contents of the screen. PA2 can be useful if you have pressed the ERASE INPUT or CLEAR key accidentally or have typed unwanted information but not yet pressed the Enter key or a function key.

How PA Keys Affect ISPF Operation

Generally, PA1 is used to terminate TSO commands or CLISTs running under ISPF. However, some TSO commands and CLISTs process PA1 in their own way.

Notes:

1. If you are running in GUI mode, you must switch back to your host session to process PA1.
2. If you are running in GUI mode, you must press Alt-Home (that is, hold down the Alt key and press the Home key) to process PA2 from the GUI display.

Restrictions that apply to CLIST attention exits are described in the *ISPF Dialog Developer's Guide and Reference* Also, ISPF should not be started from a CLIST that contains an attention exit because results are unpredictable.

If PA1 is pressed while ISPF is in full-screen mode after the keyboard has been unlocked, it is treated as a RESHOW request. If PA1 is pressed again, the current function is terminated and either the primary option menu or a top-level selection panel supplied by the dialog developer is displayed.

When an ISPF function is running, if the RESET key is pressed to unlock the keyboard and PA1 is pressed, ISPF attempts to terminate the current function and redisplay the primary option menu. The attempt might not always be successful; for example, if there is an error in MVS allocation, the attempt fails. A failure might cause unpredictable results such as waits, loops, abends, or incorrect and unrelated error messages.

AUTOTYPE - Automatic Data Set Name and Member Name Completion

The AUTOTYPE function is not available on all ISPF panels. It works only on panels that are specifically written to understand it.

If you assign the value of AUTOTYPE to a program function key, you can type a partial data set or member name into a library, member or data set name field, press the function key, and let ISPF complete the name for you.

AUTOTYPE automatically searches the catalog or PDS directory to find names that match what you entered. You can even type a pattern to limit the names that AUTOTYPE will return. AUTOTYPE works only on panels that have been enabled to use the function. You can also enable your own applications to use AUTOTYPE (see "Enabling Applications to Use AUTOTYPE" on page 126).

If you are using a terminal emulator, you can assign an easily reachable key to the function key that invokes AUTOTYPE. For example, you can use a control key combination or any other key or combination that is within easy reach. You can also use the function key directly.

Within ISPF, AUTOTYPE is enabled for the following panels:

- Edit, Browse, and View (options 1 and 2, including recursive edit/browse/view, copy, replace and move panels).
- Library Utility panels (option 3.1)
- Data Set Utility panels (option 3.2, including Rename)
- Move/Copy (option 3.3)
- Data Set List entry panel (options 3.4)
- Reset ISPF Statistics (option 3.5)
- Hardcopy Utility (option 3.6)

- Download/Upload Data Set To/From Workstation (options 3.7.2)
- SuperC Compare (options 3.12 and 3.13 in all fields)
- SuperC Search (options 3.14 and 3.15, all fields)
- SCLM View and Edit (options 10.1 and 10.2)
- SCLM Library Utility (option 10.3.1)
- Sublibrary Management (option 10.3.2)
- SCLM Migration Utility (option 10.3.3)
- SCLM Delete Group Utility (option 10.3.9)
- SCLM Build and Promote (options 10.4 and 10.5)
- Preprocessed panel utility (ISPPREP)
- Dialog tag language compiler (ISPDTLC)

On panels that are not enabled for AUTOTYPE, pressing the AUTOTYPE key is the same as pressing ENTER. The variable ZVERB is set to the value 'AUTOTYPE'.

How to Use AUTOTYPE

1. Type a partial name (zero or more characters) into a **Library** field (project, group, type, or member) or the **Other Data Set Name** field.
2. Press program function key that has been set to AUTOTYPE. ISPF sets the field to the correct value.
3. If you immediately press the program function key again, ISPF retrieves the next data set or member name. Up to 100 data set names and 700 member names can be retrieved.

Rules for Specifying the 'Other Data Set Name' Field

The prefix or pattern you specify is that which is to the left of the cursor. For example, if the field is 'CLIST(ABC)', with the cursor under the letter 'S', then the pattern used is 'CLI'. For more information, see "Cursor Position Sensitivity" on page 126.

- If the content of the **Data Set Name** field does not begin with a quote, your TSO prefix is added. If the field does begin with a quote, no prefix is added. Unquoted data set names are not processed if you do not have a TSO prefix (except for on the Data Set List Utility panel).
- The pattern can be any pattern similar to what you use in your data set list (except that you don't have to specify your TSP prefix as described above). Trailing wildcards are automatically added. For example, in a **Data Set Name** field, typing 'CHR' will result in a pattern of *prefix*.CHR*.** when searching the catalog.
- If you include a left parenthesis, followed optionally by a member name or pattern, the data set is assumed to be a PDS and the member name is returned. For example 'CLIST(XY' would search *prefix*.CLIST for members matching the pattern *XY**. A trailing * is always added to the member name to create a pattern.

Rules for Library Fields - Project, Group, Type, Member

- If the cursor is in the **Project** field, the other fields are not used as part of the data set name search and are erased.
- If the cursor is in a **Group** field, the project name and type name (if any) are added to create the search pattern. Each group name is used only once, even if there are many types in that group. As each group name is displayed, the first

type name found for that project.group combination is also retrieved. The member name, if any, is not used and is erased.

- If the cursor is in the **Type** field, the project and first group name are used to create the pattern. The **Type** field is updated and the member name is erased.
- If the cursor is in the **Member** field, the project, first group, and type are used as the data set name. The second, third, and fourth groups are not used.

Cursor Position Sensitivity

The pattern or prefix used to search for names is only that which is to the left of the cursor.

In this way you can refine your search simply by moving the cursor. For example; suppose you have one hundred data sets called 'SYS1.A234.RGG.*' and you plan to use one named 'SYS1.A234.RGG.DBD0223.L422.FEB0299.TERRA'. You could type 'SYS1.A234.RGG' and press the AUTOTYPE key. That might return 'SYS1.A234.RGG.DBD0211.X331.AUG0599.FIRMA'. You can refine the next value returned by typing a '2' over the first '1' in 'DB0211' and then press the AUTOTYPE key again. This will use the new pattern and get you closer to the desired value.'

Restrictions

AUTOTYPE only retrieves cataloged alias names. It will not retrieve generation data group or generation data set names. AUTOTYPE does not use the **Volume** field on any panel. It sets the **Volume** field associated with the current data-set name field to blank.

AUTOTYPE does not use any ISPF name change exits or data set list retrieval exits.

AUTOTYPE retrieves a maximum or 100 data sets and 700 member names before cycling through the list from the beginning.

Enabling Applications to Use AUTOTYPE

You can enable applications to use AUTOTYPE by making some minor panel modifications as follows:

1. At the beginning of the)REINIT section, add the following lines. Make sure the subsequent line in the)REINIT section starts in column 1 so that it does not become part of the IF clause you have inserted.

   ```
   IF (&ZNXTMSG=NEXT  .CSRPOS = &ZCSRP
                      .CURSOR = &ZCSRV
   ```

2. At the end of the)REINIT section add the following line in column 1.

   ```
   REFRESH (*)
   ```

3. At the beginning of the)PROC section add the following lines. Make sure the subsequent line in the)PROC section starts in column 1 so that it does not become part of the IF clause you have inserted. If there is a line that says: `.RET = OFF` in the)PROC section, it should go before these lines.

   ```
   &ZCSRV = .CURSOR
   &ZCSRP = .CSRPOS
   &NAMES='ZCSRV ZCSRP PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN ZCMD
   PANEXIT ( (NAMES) , LOAD,ISRAUTOT)
   IF (&ZNXTMSG='ISRT') EXIT
   ```

Modify the line that assigns the variable &NAMES. This assignment contains a list of variable names on the panel. They must all be specified in order. Use an asterisk (*) for names that are not relevant for your panel.

The values in the &NAMES variable are:

1. The variable containing the Cursor field name
2. The variable containing the cursor offset
3. The name of the Project variable on the panel
4. The name of the first Group variable.
5. The name of the second Group variable
6. The name of the third Group variable
7. The name of the fourth Group variable
8. The name of the Type variable
9. The name of the Member variable
10. The name of the Other Data Set Name variable
11. The name of the command line variable (clears the command line)

The cursor and name variables (described here as ZCSRV, ZCSRP, and NAMES) can have any names you choose but they must match the names used in the)INIT section and the PANEXIT statement in the)PROC section.

If a name references a read-only field, add a dash to the end of the name. If you need to limit the size of the returned name, you can append the maximum length, after a period, to the name; for example, `ODSN.44`. You can disable member searches for a data set name field by adding a percent sign to the end of the field name; for example, `ODSN.44%` or `ODSN%`.

Panels defined in dialog tag language (DTL) can be enabled for AUTOTYPE through DTL keywords. See the *Developer's Dialog Reference* manual for more information.

How PA Keys Affect ISPF Operation

Chapter 5. Libraries and Data Sets

An ISPF library is a cataloged partitioned data set or a partitioned data set extended (PDSE).The PDSE is a data set type introduced in the Data Facility Product Version 3.2 (DFP V3.2). See "Partitioned Data Set Extended (PDSE)" on page 152 for more information.

The ISPF library has a three-level name consisting of a project, group, and type. An optional library member name can also be included. A member consists of programming code, data, or text.

ISPF displays library names on line 1 of a data display, such as the member list shown in Figure 41 on page 137. Each library generally contains members with the same type of information.

Before you can create and use a new ISPF library or data set, you must allocate it using option A of the Data Set utility (option 3.2). The following section explains how to name the ISPF library or data set. See the **Data Set Utility (Option 3.2)** section of the *ISPF User's Guide Volume II* for step-by-step instructions on allocating data sets.

Naming ISPF Libraries and Data Sets

On data entry panels that require a library or other data set name, such as the Data Set Utility (option 3.2) and Edit Entry (option 2) panels, two groups of fields are provided: one for entering an ISPF library name and one for entering another partitioned or sequential data set name. The next two sections explain these two groups of fields.

ISPF Library Names

To name an ISPF library, you must specify at least a project, group, and type. For example:

```
Project  . . ISPFPROJ
Group  . . . TEST
Type . . . . PLI
```

Project
The common identifier for all ISPF libraries belonging to the same programming project. This name must be your user ID unless you are using a specific project name that has been predefined in the MVS master catalog.

Group The identifier for a particular set of ISPF libraries, that is, the level of the libraries within the library hierarchy. For example, the group name of your private library could be PRIVATE or perhaps your first name, such as Joe in the example in Figure 39 on page 133.

Type The identifier for the type of information in the ISPF library, such as PL/I, SCRIPT, or PANELS.

Standard ISPF Naming Conventions

Each component of the library name can be up to 8 alphanumeric or national characters; the first one must be alphabetic. This conforms to standard TSO data set naming conventions. For convenience, any cataloged data set (sequential or

partitioned) with a three-level name can be entered in the **Project**, **Group**, and **Type** fields, with one level of the name in each field. If a cataloged data set with four or more levels is to be entered, multiple levels of the name may be entered in each field, with each level being separated by a period.

If both a library and a data set name are specified on the same panel, the data set name takes priority. Therefore, to specify a library, leave the **Data Set Name** field blank.

When the library identification appears in a title line or message, the project name, group name, and type name are separated with periods. A member name, if applicable, is enclosed in parentheses. For example:

```
PROJECT.GROUP.TYPE(MEMBER)
```

On most data entry panels that allow ISPF library name specification, a **Member** field is available:

```
Member  . . . PROJ1
```

Member
The name of an ISPF library or other partitioned data set member. Leaving this field blank or entering a pattern causes ISPF to display a member list. See "Displaying Member Lists" on page 135 for more information.

A *pattern* is a partial member name that uses either or both of the following symbols as place holders: an asterisk

An asterisk represents a string of characters; a percent sign represents only 1 character. ISPF matches the pattern to any like members in the specified data set.

The ISPF library's project, group, and type must always accompany the member name, if entered. If you try to edit a member that does not exist, ISPF provides an Edit display screen with a blank data area.

Member names entered in the **Member** field or those enclosed in parentheses and entered in the **Data Set Name** field must follow standard ISPF naming conventions.

If you have a partitioned data set with members whose names do not follow ISPF naming conventions, ISPF allows limited processing, as follows:

- View (option 1) allows any character string as a member name in either the **Member** or **Data Set Name** field and attempts to View or Browse the specified member.
- Edit (option 2) allows an *existing* member with a nonstandard member name to be edited. You cannot create a member with a nonstandard member name.

ISPF cannot process member names that begin with a blank or have embedded blanks which can cause unpredictable results. Also, ISPF cannot process member names that include special characters, such as an ampersand (&). CLIST processing in both Foreground (option 4) and Batch (option 5) can result in a run-time error.

Other Partitioned or Sequential Data Set Names

You can use the following field to specify any partitioned or sequential data set:

```
Other Partitioned or Sequential Data Set:
   Data Set Name . . .
```

Data Set Name
Any fully qualified data set name, such as:

```
Data Set Name . . . 'USERID.SYS1.ASM'
```

You can include either a TSO user prefix or user ID as the first-level qualifier of the data set name. If you omit the single quotes and if you have created a TSO user prefix, that prefix is automatically added to the beginning of the data set name. If you omit the single quotes and if you do not have a TSO user prefix, no prefix is added, and the name is used exactly as it appears.

If you include your user prefix or user ID, enclose the data set name with apostrophes. If you include the apostrophe at the beginning of the data set name but omit the one at the end, ISPF inserts it for you.

Note: ISPF does not support multivolume data sets or partitioned data sets with record format FBS or VBS.

For partitioned data sets, a member name enclosed in parentheses can follow the data set name. For example:

```
Data Set Name . . . 'SYS1.PROCLIB(ASMHC)'
```

When you omit the member name and parentheses or use a pattern ISPF displays a member list. See "Displaying Member Lists" on page 135 for more information.

You can refer to generation data sets by using a signed or unsigned number in place of a member name in the **Data Set Name** field only. For example:

```
Data Set Name . . . 'gds.test(0)'
```

This example refers to the most recently allocated data set in the generation data group. Minus numbers refer to previously allocated data sets; positive refer to unallocated.

Note: For Edit, Browse, and View, a VSAM data set can be specified if the ISPF Configuration Table enables VSAM processing.

Volume Serials

Along with a data set name, you can optionally specify a volume serial. If you do, the system catalog is not used. For example:

```
Volume Serial . . . _____    (If not cataloged)
```

Volume Serial
A real DASD volume or a virtual volume residing on an IBM 3850 Mass Storage System. To access 3850 virtual volumes, you need MOUNT authority, which is acquired through the TSO ACCOUNT command or the RACF TSO AUTH CLASS command.

Library Concatenation

Whenever the first **Group** field is accompanied by three additional fields horizontally across the screen, you can enter a *library concatenation sequence*, which is a series of group names chained together. ISPF searches these groups in the sequence that you enter them.

You can concatenate libraries of the same type, but only libraries that belong to the same project. You will usually concatenate the lowest-level library ahead of the

next higher-level library, and so on, in bottom-to-top order. Therefore, concatenation is usually most effective if this search sequence is the same as the library hierarchy.

For example, new library members or members undergoing changes generally reside in libraries used by program developers. A test library may contain members that have been unit tested and are ready for integration test. A master library might contain fully tested members that correspond to a previously released version of the program.

Concatenated libraries must have consistent record formats and logical record lengths. You can use concatenation with the following ISPF functions:

- Viewing
- Browsing
- Editing
- Selecting Library Utility (option 3.1) functions:
  - Print index or complete data set
  - Browse, delete, edit, print, rename, or view members
  - Compress data set.
- Copying data sets or members
- Compiling
- Assembling
- Link editing
- SCRIPT/VS processing.

Note: You can also use additional input libraries for compilations and assemblies.

Figure 39 on page 133 shows a sample three-level hierarchy consisting of a set of master libraries, a set of test libraries, and three sets of private development libraries identified by user ID. Using this hierarchy, a typical concatenation sequence for a project of ISPFPROJ, a type of DATA, and a member PGM1 is:

```
ISPF Library:
   Project . . . ISPFPROJ
   Group . . . . JOE      . . . TEST____ . . . MASTER__
   Type  . . . . DATA
   Member  . . . PGM1____
```

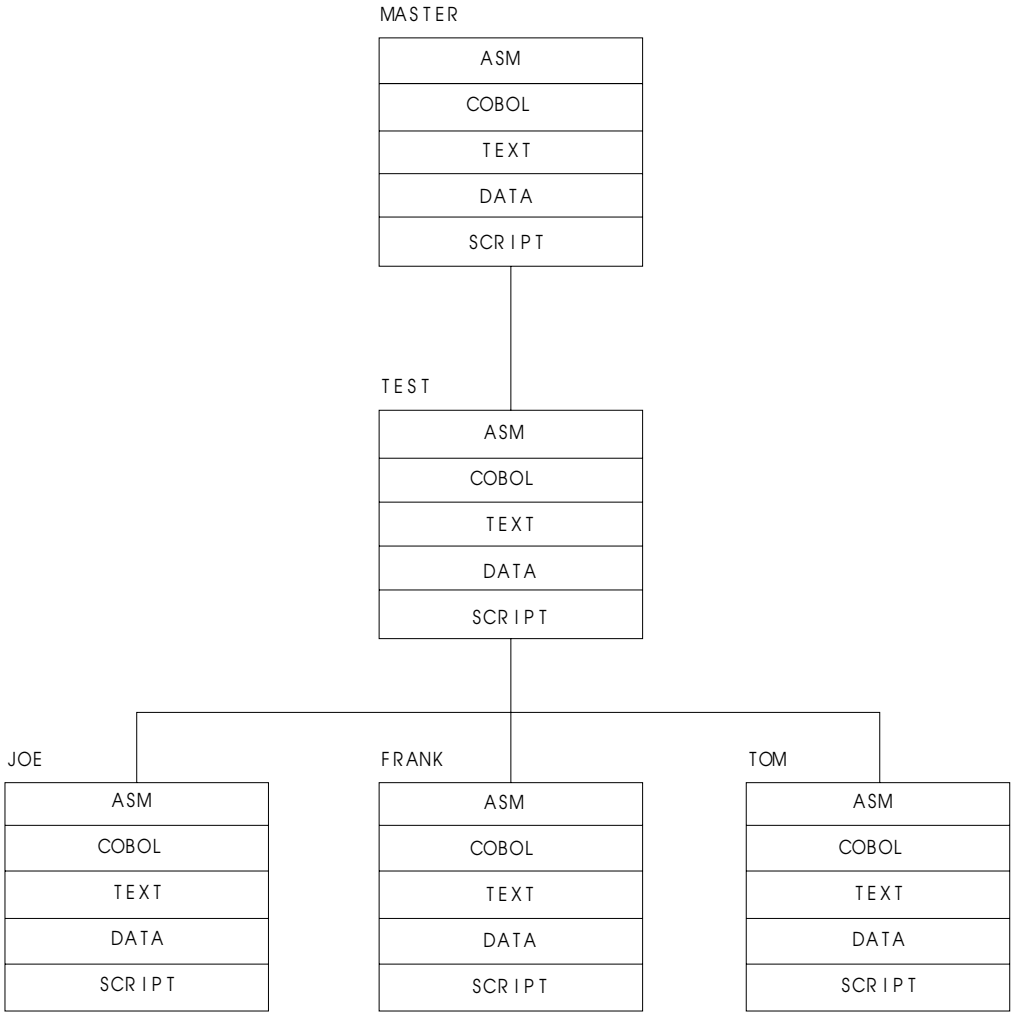


Figure 39. Hierarchy of ISPF Libraries

In this example, the search for member PGM1 goes through libraries:

```
ISPFPROJ.JOE.DATA
ISPFPROJ.TEST.DATA
ISPFPROJ.MASTER.DATA
```

Concatenation during Editing

Using concatenation during editing provides a way to copy members to your development library. Use the concatenation sequence to search the libraries for the member to edit. The edited member is saved in your development library, the first library in the concatenation sequence, while the unchanged version remains in the test or master library. When the new version is fully tested, you can use the Move/Copy utility (option 3.3) to:

- Move the new version to a higher-level library.

Concatenation during Language Processing

The purpose of concatenation during language processing is to:

- Help you include source segments in their proper order when using INCLUDE or COPY statements or when using SCRIPT imbed controls
- Allow debugging of new or changed programs without altering the contents of the test or master libraries.

Library Concatenation

The output from a compilation or assembly (an object module) or from a link edit (a load module) is stored in the lowest-level OBJ or LOAD library, the first library in the concatenation sequence.

Using Member Selection Lists

A *member selection list*, also called a member list, is initially an alphabetic list of the members of an ISPF library or TSO partitioned data set. Table 5 provides a quick reference to the primary options that display member lists and their differences. In the Type of Selection column, *single* means that ISPF processes only the line command that is the closest to the top of the list, ignoring all others. *Multiple* means that you can enter more than one line command simultaneously. The numbers in parentheses refer to notes following the table. See "Member Selection List Commands" on page 142 for more information about the line commands shown in the table.

Table 5. Member Selection List Differences

Primary Options	Type of Selection	Valid Line Commands	Prompt Field Available
View (1)	Single	S,V (4)	No
Browse (1)	Single	S,B (4)	No
Edit (2)	Single	S,E (4)	No
Library (3.1)	Multiple	B,C,D,E,G,M,P,R T,V,W	Yes
Move/Copy (3.3)	Multiple	B,S (1)	Yes
Data Set List (3.4)	Multiple	B,C,D,E,G,M,P,R T,V,W (2)	Yes
Reset (3.5)	Multiple	S	No
Convert (3.10)	Multiple	S	Yes
SuperC (3.12)	Multiple	S	No (3)
SuperCE (3.13)	Multiple	S	No (3)
Search-For (3.14)	Multiple	S	No
Foreground (4)	Single	S	No
Batch (5)	Single	S	No
Workplace (11)	Multiple	B,C,D,E,G,P MO,R,S,T,V,W	No

Notes:

1. For the Move/Copy utility, B (browse member) enables you to browse members of an ISPF library or another partitioned data set before moving or copying them without having to use browse on another panel. Then, use S (select) to select the member or members to move or copy. See "Line Commands for the Move/Copy Utility" on page 147 for more information.
2. When you select M (display member list) line command on a data set list, you can use B (browse member), D (delete member), E (edit member), P (print member), R (rename member), and V (view member). You can also enter TSO commands, CLISTs, and REXX EXECs.

   S (select) is valid also, but only when the B, CO, E, MO, RS, or V line commands are used on a data set list.
3. Instead of a **Prompt** field, this member list has an **OLDMEM** field, which you can use to enter the name of a member in the old data set. See the **SuperC**

Member Lists section of the ***SuperC Utility (Option 3.12)*** chapter of the *ISPF User's Guide Volume II* for more information about this field.

4. For your convenience ISPF supports E as a select character from Edit member lists in addition to S and point-and-shoot selection. Similarly, V is supported from View member lists, and B is supported from Browse member lists.
5. When multiple members are selected from a member list that supports multiple selection and all members have been processed, the member list is scrolled such that the last member processed is positioned to the top of the member list display.

Displaying Member Lists

For each of the primary options listed in the preceding table, except Data Set List (option 3.4), you can display a member list by:

- Leaving the **Member** field blank for an ISPF library
- Omitting the member name from the name of another partitioned data set
- Entering a pattern as the member name.

  You can use a combination of asterisks and percent signs in the same pattern. However, the pattern, including the asterisks and percent signs, can contain no more than 8 characters. For example, entering the following pattern in the **Member** field:

  ```
  Member  . . . *prof___
  ```

  could display this member list:

  ```
  ISFPROF
  ISPPROF
  ISPSPROF
  ISRPROF
  LOCPROF
  SUPCPROF
  ```

When using the Data Set List utility (option 3.4), you can display a member list by:

- Entering the M (display member list) line command
- Entering the V (view), B (browse), or E (edit) line command and then using one of the methods described in the preceding list. This applies only if you are editing or browsing members of a partitioned data set.
- Entering the CO (copy) line command
- Entering the MO (move) line command
- Entering the RS (reset) line command.

On any member list, PF10 and PF11 toggle between two different views of the member list data.

Notes:

1. The column headers on a member list display (with the exception of Prompt) are point-and-shoot sort fields.
2. If you enter a slash in the line command field, the Member List Commands pop-up window shown in Figure 40 is displayed so that you can select the command you want to use.
3. The line command field is a point-and-shoot field. If you select the line command field beside a member name, the Member List Commands pop-up window shown in Figure 40 is displayed so that you can select the command you want to use.

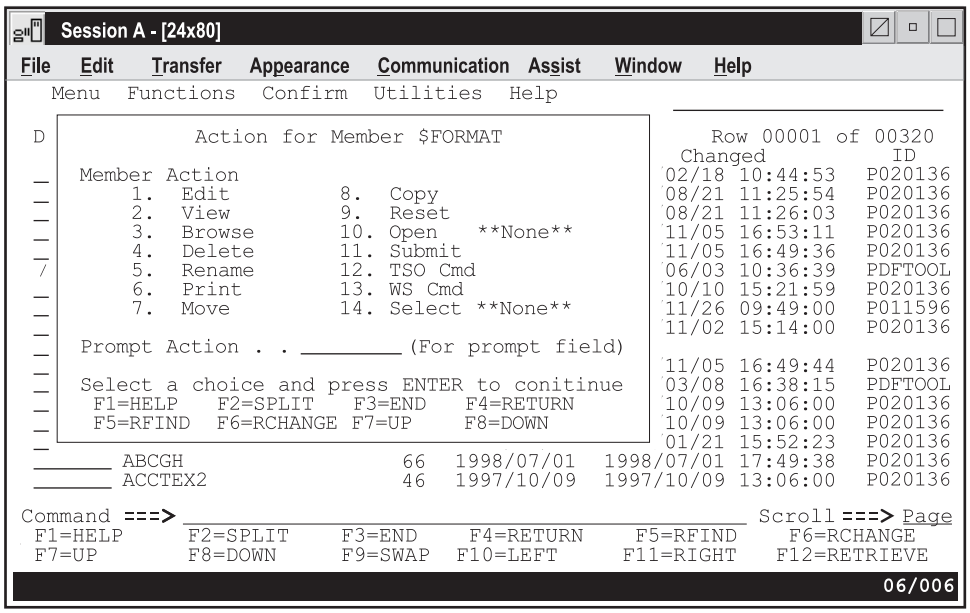


Figure 40. Member List Commands Pop-Up Window (ISRCMLEP)

Ending Member Lists

With two exceptions, you can end a member list by entering END (F15) or using = (the jump function) to go to another option. For the two exceptions, SuperC and Search-For member lists, enter RETURN (F16), CANCEL, or =. On these member lists, the END command processes your selections.

ISPF Member Statistics

On member lists, column headings appear in the national language. The information shown under the column headings contains the ISPF statistics generated for each member. You can print these statistics using option X (print index listing) of the Library utility (option 3.1) or option P (print data set list) of the Data Set List utility (option 3.4). You can also use the SAVE command to write a member list or data set list to the ISPF list data set or to a sequential data set. The statistics are displayed next to each member name.

Figure 41 on page 137 shows an example of a member list with statistics and the one-character line command field to the left of the member names.

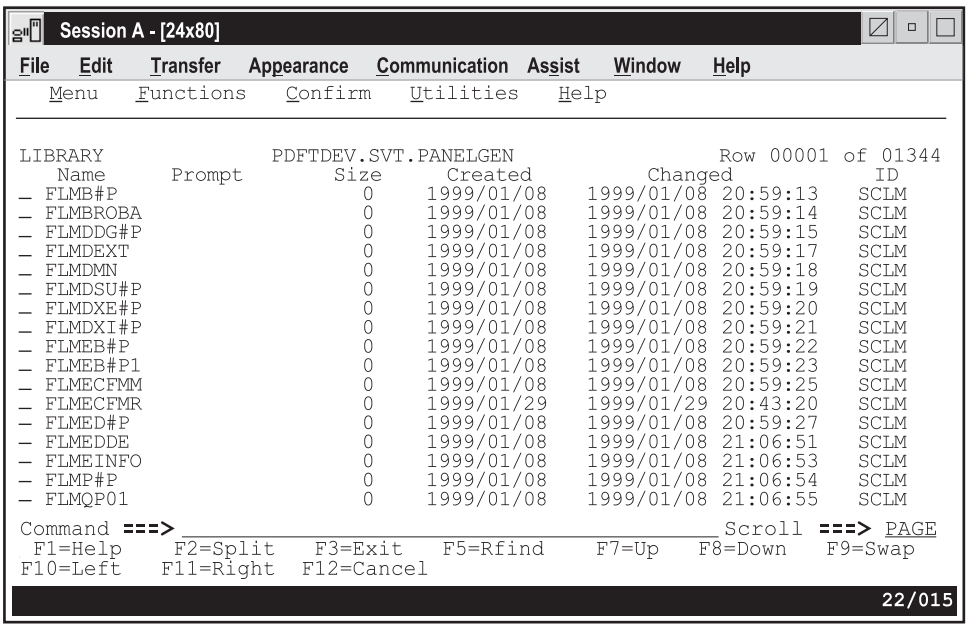


Figure 41. Member List Display (ISRUDMM)

If you want to see all of the statistics, you can scroll the screen either right or left by using PF keys 10 and 11. If you scroll right in this example, the screen looks like the one shown in Figure 42. Pressing either key repeatedly results in recycling of the screens.

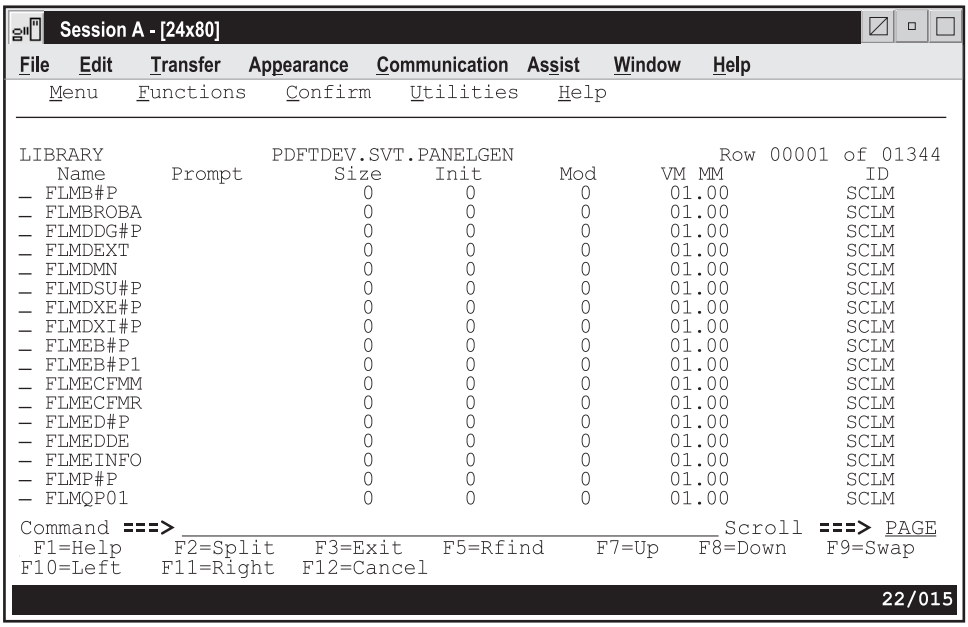


Figure 42. Member List Display (ISRUDMM)

Member List Display Panel Action Bar: The Member List Display panel action bar choices function as follows:

Menu See "Menu Action Bar Choice" on page 7 for information on the Menu pull-down.

Functions

The Functions pull-down offers you the following choices:

1 **Save List**. Saves the member list into the list data set.

2 **Change Colors...** Displays the Member List Color Change Utility panel (Figure 43) to allow you to change one or more of the Member List Field Attributes and press Enter to see the effect immediately. Clearing a field or selecting the **Defaults** field restores defaults.

Note: You can also display this panel by typing `MLC` on the Command line and pressing Enter.

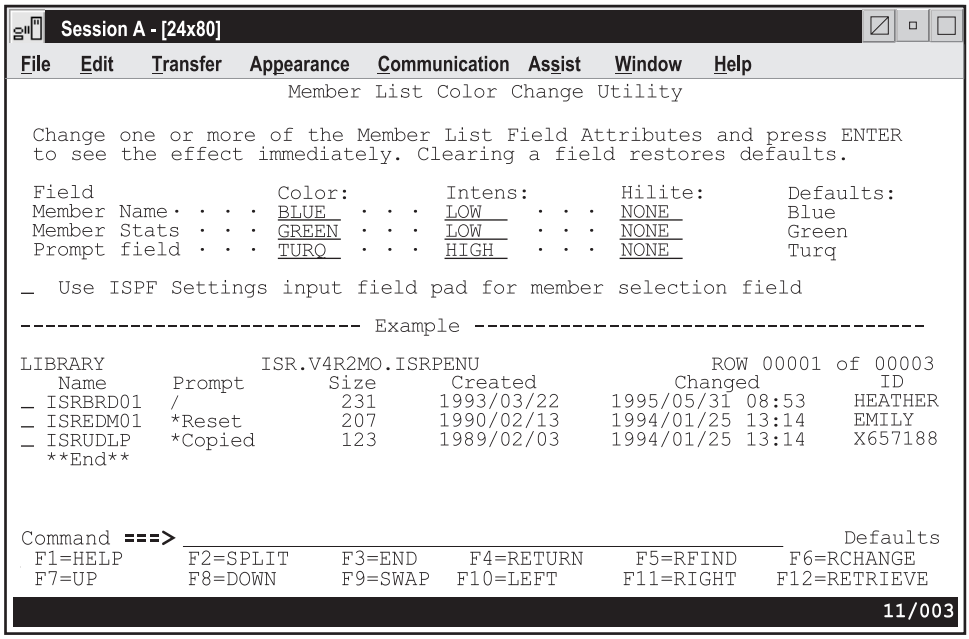


Figure 43. Member List Color Change Utility Panel (ISRMLCP)

3 **Initial Sort View...** Displays the Member List Sort Field View panel (Figure 44 on page 139) to enable you to select the member list field to be sorted on, prior to the display of an enhanced member list. Member Name is the default View. Some other sort views that you can choose are:

- RECFM=BLK sets the initial view for blocked (FB,VB,...) data set formats.
- RECFM=U sets the initial view for unformated (Load,...) data set formats.
- Any of the following conditions result in a default sort view on member name.
  - Library field selected as initial sort view and member list is based on a single data set.
  - Created or Alias selected as initial sort view and extended command member list.
  - A member name is given as input to member list action.

Note: You can also display this panel by typing `MLS` on the Command line and pressing Enter.

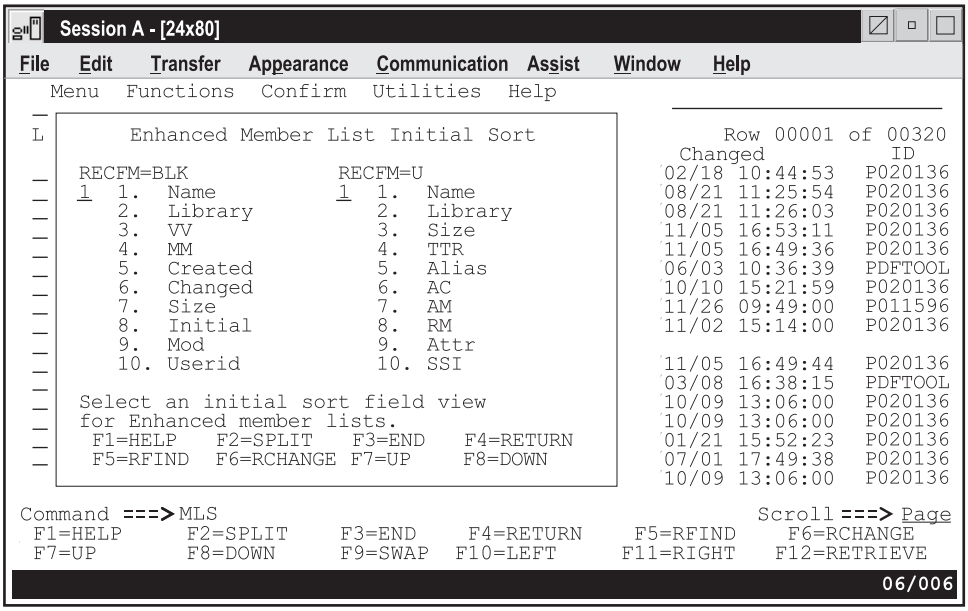


Figure 44. Member List Sort Field View Panel (ISRMLIS)

Note: When using LLA in FREEZE mode, member lists might appear to be out of SORT order because ISPF uses direct reads of the data set directory for initial member list build and SORT, and uses BLDL for the displayed statistics.

Confirm

Select 1 or 2 to set delete confirmation ON or OFF.

Utilities

See "Utilities Action Bar Choice" on page 7 for information on the Utilities pull-down.

Help The Help pull-down offers you the following choices:

1 Workplace General
2 Library/DSLIST General
3 Scrolling
4 Pattern matching
5 LOCATE command
6 SORT command
7 SAVE command
8 RESET command
9 SELECT command
10 MLC command (member list color)
11 MLS command (member list sort)
12 S line command
13 Statistics
14 Appendices
15 Index

Member List Display Panel Fields: ISPF generates statistics each time you edit a member, unless your edit profile is set to `STATS OFF`. The following fields identify the statistics in a member list:

Using Member Selection Lists

Note: The column headers on a member list display (with the exception of Prompt) are point-and-shoot sort fields.

Name Name of the member.

Prompt
The **Prompt** field serves a variety of purposes. You can rename a member by typing the new name to the right of the member name. You can type a slash character (/) in the first position of the Prompt field so you can define additional behaviors for a given action.

Also, the Prompt field acts as a status field, showing information about the last action taken for a member. If you run an edit macro or TSO command against a member, the 7-character informational status that is returned in the dialog variable ZPROMPT at the completion of the service is shown in this field.

Lib Library number. The **Lib** field appears only if you specify a concatenated sequence of libraries. It shows the library that contains the member. In this example, if the member resides in the second library in the sequence, a `2` appears in the **Lib** field.

VV.MM
Version number and modification level. The version number is set to 1 and the modification level is set to 0 when the member is created. The modification level is the number of times this version has been modified. For example, `02.15` means version 2, modification 15.

If a member name is just an alternate name for another member, ALIAS appears in this field.

Created
Date this version was created. The format used depends on your national format. For example, `90/06/27` means June 27, 1990 to some, but so does `06/27/90` and `27/06/90` mean it for others.

Changed
Date and time this version was last modified; date is shown in the national format (see Created). Time is shown using a 24-hour format. For example, `17:20` means 5:20 p.m..

Size Current number of lines. The largest number this field can display is 65,535.

Init Number of lines when the member was first saved. The largest number this field can display is 65,535.

Mod Number of lines in the current member that have been added or changed. If the data is unnumbered, this number is zero. The largest number this field can display is 65,535.

ID The user ID of the person who created or last updated this version.

When you use View, Browse, and Edit, the current version and modification level are displayed in the title area, line 1, following the library and member name. You can change the version number, the user ID, or both, with the Reset ISPF Statistics utility (option 3.5) or with the LEVEL and VERSION Edit primary commands. Changing the version number updates most of the other statistics.

If you use the ISPF editor to delete all lines in a member of an ISPF library and then save the member, the statistics show that the member still exists but has a length of zero. To delete a member, including its statistics, use the Library utility (3.1).

Member lists displayed when you use the Data Set utility (option 3.4) contain an extended line command area and do not display the created date.

Load Module Library Member Statistics

Figure 45 shows that the ISPF library statistics displayed in a member list have a different format for load module libraries. See "Member List Display Panel Action Bar" on page 137 for a description of the action bar choices on this panel.

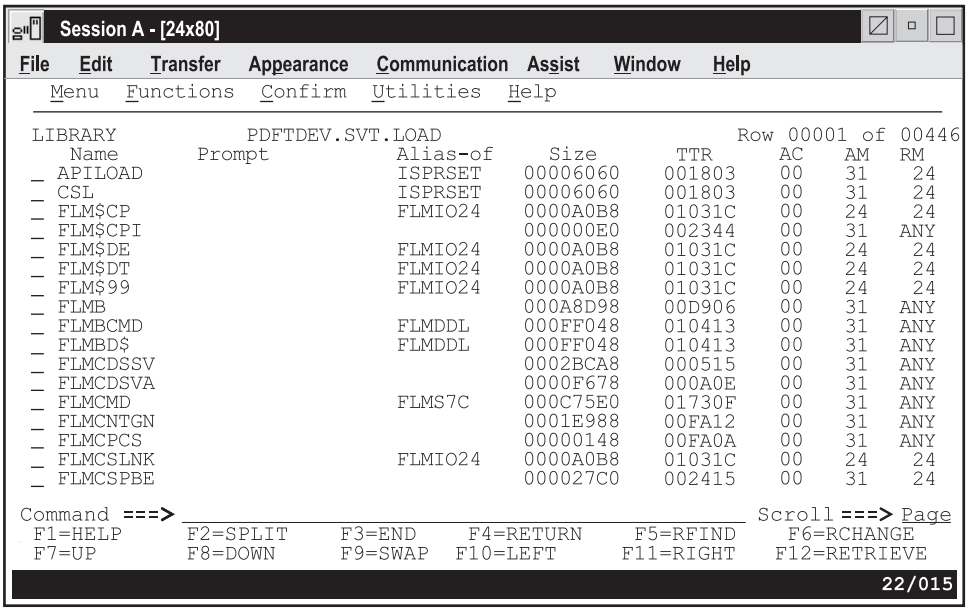


Figure 45. Load Module Library Display (ISRUDMM)

If you want to see all of the statistics, you can scroll the screen either right or left by using PF keys 10 and 11. Pressing either key repeatedly results in recycling of the screens.

The fields on a member list display for a load module library are:

Note: The column headers on a member list display (with the exception of Prompt) are point-and-shoot sort fields.

Name Name of the member.

Prompt The **Prompt** field serves a variety of purposes. You can rename a member by typing the new name to the right of the member name. You can type a slash character (/) in the first position of the Prompt field so you can define additional behaviors for a given action. Also, the Prompt field acts as a status field, showing information about the last action taken for a member.

Lib Library number. The **Lib** field appears only if you specify a concatenated sequence of libraries. It shows the library that contains the member. In this example, the member resides in the second library in the sequence.

Size Size of the member in hexadecimal. The largest number this field can display is 'FFFFFFFF'X.

TTR Relative block address.

Alias-of Name of the member for which this member is an alias. See following note.

AC Authorization code.

AM Addressing mode.

RM Residency mode.

Attributes The member's attributes:
- **NX** Not executable
- **DATA** Can be loaded only
- **OVLY** In overlay structure
- **RF** Refreshable
- **RN** Can be reentered
- **RU** Reusable
- **TEST** Module to be tested.

SSI System Status Index

Note: Question marks (?) are placed in the member list of a load module library for members that have load module directory fields that are not valid. For example, module 14, shown in Figure 45 on page 141, contains alias and authorization code information that is not valid.

Load module library lists displayed using the Data Set List utility (option 3.4) contain an extended line command area and do not display the created date.

Member Selection List Commands

If the member list is too large for the screen, you can see other parts of the list by using the UP and DOWN scroll commands. These commands are valid for all member list displays. However, because a member list display can be no wider than 80 characters, you cannot use the LEFT and RIGHT scroll commands. Refer to the *ISPF Dialog Developer's Guide and Reference* for more information about scroll commands.

The following *primary* commands are valid for all member list displays. You enter these commands on the Command line:

- CONFIRM
- FIND
- LOCATE *string*
- RESET
- RFIND
- SAVE [*list-id*]
- SELECT *pattern* [*lcmd*]
- SORT [*field1*[*field2*]]
- MLC
- MLS
- REFRESH

The following *line* commands can be used with member lists. These are one-character commands that are entered to the left of the member name. The option you are using determines:

- Whether you can enter more than one line command simultaneously
- Which line commands are valid

- Whether a **Prompt** or **OLDMEM** field is available.

Table 5 on page 134 provides a quick reference to the differences between member lists and the line commands available on each one. The line commands are:
- B (browse member)
- C (copy member)
- D (delete member)
- E (edit member)
- G (reset member)
- M (move member)
- P (print member)
- R (rename member)
- S (select member)
- T (TSO command)
- V (view member)
- W (WS command)
- = (repeat last command).

The S line command is available for all member list displays except the Library and Data Set List utilities. See "S Line Command" on page 146 for more information.

The B line command is available only for the Library, Move/Copy, and Data Set List utilities. The D, E, P, R, and V line commands are available only for the Library and Data Set List utilities.

Note: For the Data Set List utility, these line commands are valid only after you enter the M (display member list) line command. See "Library and Data Set List Utility Line Commands" on page 148 for information.

ISPF ignores any unprocessed member list commands when you leave a member list.

Primary Commands

Confirming a Delete Command: The CONFIRM primary command controls display of the Confirm Delete panel. Use the following format:

CONFIRM [ON | OFF]

You can use the following operands with the CONFIRM command:

ON Tells ISPF to display the Confirm Delete panel when you enter the D (delete data set) line command or TSO DELETE command.This is the default setting.

OFF Tells ISPF not to display the Confirm Delete panel.

For example, the following command would tell ISPF not to display the Confirm Delete panel:

```
CONFIRM OFF
```

Note: Confirm is forced on from the workplace member list with a default action of "D".

Find a Character String: To find a character string within the specified field use the FIND or the RFIND command. Only one of the fields on the member list can be specified at a time for the search. Use the following format:

FIND string field [NEXT]

[[ALL] [PREfix]

[[FIRST] [SUFfix]

[[LAST] [WORD]

[[PREV]

NAME is the default field. NEXT is the default operand. For example, the following command tells ISPF to find the last occurrence of the character string XLC in the NAME field.

```
FIND XLC NAME LAST
```

ISPF automatically scrolls to bring the line containing the character string to the top of the list.

Use the RFIND command to repeat the search without reentering the character string.

Locating a Data String: To find a data string, you can enter a LOCATE command in the **Command** field on any member list display. The format of the command is:

LOCATE *string*

where:

string
A data string that is used to find an entry based on how the member list is sorted.

ISPF searches the field by which the member list was sorted for an entry equal to *string*. Either the entry, if found, or else the entry that immediately precedes the entry that you are searching for is scrolled to the top of the list.

For example, if the member list shown in Figure 45 on page 141 is sorted by name, the following command causes member MODULE12 to scroll to the top of the list:

```
LOCATE MODULE12
```

Removing Unwanted Line Commands and Messages: The RESET command removes unprocessed line commands and messages that show the result of line command processing. This command has no operands.

Writing a Member List to a Sequential Data Set: The SAVE primary command writes a member selection list to a sequential data set. The format of the SAVE command is:

SAVE [*list-id***]**

where:

list-id
Optional. A user-specified qualifier of the sequential data set to which the member list is written.

ISPF names the data set:

```
prefix.userid.list-id.MEMBERS
```

prefix Your data set prefix, as specified in your TSO user profile. If you have no prefix set, or if your prefix is the same as your user ID, the prefix is omitted and the data set name is userid.list-id.MEMBERS.

userid Your TSO user ID.

The data set is created if it does not exist, or written over if it exists and has compatible attributes.

ISPF writes the member list in the current sort order and as it appears on the display, except for the column headings, Line Command fields, and anything you have typed on the display.

If you omit the list ID, ISPF writes the member selection list in the current sort order, including column headings, to the ISPF list data set. Processing is the same as using option X of the Library utility (primary option 3.1), except that data set information is not printed.

Selecting a Member: You can use the SELECT, or S, command as either a primary command or a line command.

SELECT Primary Command: The SELECT primary command allows you to select one or more members in a member list, whether they are displayed or not. When you enter it on a member list displayed using the Edit option, this command even creates a member if you specify the complete member name of a member that does not exist.

The SELECT command optionally provides a quick method of calling the same line command for one or more members. The format of the SELECT command is:

SELECT {*pattern* | * }[*lcmd*]

where:

pattern
Either a complete member name or a partial member name that contains one or more asterisks (*), percent signs (%), or both as place holders. See "Displaying Member Lists" on page 135 for more information about using patterns.

***** An asterisk, which means you want to select all members in a member list.

lcmd
One of the following optional line commands: S (select), B (browse), V (view), D (delete), E (edit), or P (print). On a member list that has an expanded line command area, such as one generated by the M (member list) line command in the Data Set List utility (option 3.4), you can also enter a TSO command, CLIST, or REXX EXEC. If you do not enter a line command, S is the default.

The member list shown in Figure 41 on page 137 contains members INT and INTTOOL. The following command selects these members for printing:

```
SELECT INT* P
```

S Line Command: You can enter the S line command at the beginning of a line, ahead of one or more member names. For example, in Figure 45 on page 141, you could select member MODULE9 by moving the cursor to the left of the member name, typing S, and pressing Enter.

Note: On member lists displayed with the View, Browse, Edit, Foreground, and Batch options, ISPF processes only the first S entered, ignoring all others.

With the Move/Copy utility and the Convert utility, you can rename members by entering new member names in the **Prompt** field to the right of the member name.

Sorting a Member List: The SORT primary command arranges a member list according to the fields you specify. The sort sequence, ascending or descending, is determined by the fields you choose and is maintained between member list displays.

The format of this command is:

SORT [*field1***[***field2***]]**

where:

field1
The primary field by which the member list is sorted.

field2
The secondary field by which the member list is sorted.

Table 6 and Table 7 show:
- Valid values for *field1* and *field2*
- The sort sequence used for each field
- A description of each field name.

Table 6. Sort Fields for Source Libraries

Field	Sequence	Description
Name	Ascending	Member name
Lib	Ascending	Library in concatenation sequence
VV	Ascending	ISPF version number
MM	Ascending	ISPF modification level
Created	Descending	Creation date
Changed	Descending	Date and time last changed
Size	Descending	Current number of records
Init	Descending	Initial number of records
Mod	Descending	Number of modified records
ID	Ascending	Last user

Table 7. Sort Fields for Load Libraries

Field	Sequence	Description
Name	Ascending	Member name
Lib	Ascending	Library in concatenation sequence

Table 7. Sort Fields for Load Libraries (continued)

Field	Sequence	Description
Size	Descending	Load module size
TTR	Ascending	TTRN of beginning of load module
Alias-Of	Ascending	Member this is an alias of
AC	Ascending	Authorization code
AM	Descending [2]	Addressing mode
RM	Descending [2]	Residency mode
Attributes	Descending	Load module attributes
SSI	Ascending	System Status Index

For example, to sort a member list by size and then by track record, you could enter:

```
SORT SIZE TTR
```

Changing Member List Field Attributes: The MLC command enables you to change one or more of the member list field attributes and to see the change immediately. Clearing a field restores the field's default setting. Use the ***Defaults*** point-and-shoot field to restore **all** field attributes to ISPF default settings.

You can also change the member selection field to use the ISPF Settings input field padding character instead of the member list field default padding character. The member list default padding character for single command selection lists is a period (.), and for multiple command selection lists it is an underscore (_).

Changing the Default Sort Order for Member Lists: The MLS command enables you to change the default sort order for all ISPF member lists. You can specify separate sort orders for Load and non-Load data sets. These sort orders apply only to ISPF Enhanced Member lists.

Refreshing the Member Lists: The REFRESH command refreshes the member list, adding new members, adding renamed members under their new names, and deleting members that have been removed from the list. It also resets the line command field and prompt field on the member list. Unprocessed line commands and input or messages inthe prompt fields are erased by the Refresh command.

Line Commands

Line Commands for the Move/Copy Utility: On member list displays for the Move/Copy utility (option 3.3), you can enter the following line commands at the beginning of a line, ahead of one or more member names.

B Browse the member
S Select the member.

The B (browse) line command allows you to browse a member or members to determine whether you really want to move or copy them. You can enter the B line command beside as many members as you want to. The first member that has a B line command beside it is browsed when you press Enter. When you finish

2. For the AM and RM columns, the value ANY is considered to be the largest value and will therefore sort to the top of the list.

browsing each member, the member list is redisplayed along with the unprocessed line commands. Press Enter again to browse the next member.

Once you have decided which members to move or copy, use the S (select) line command to select those members.

Library and Data Set List Utility Line Commands: On member list displays for the Library utility (option 3.1) and the Data Set List utility (option 3.4), you can enter the following line commands at the beginning of a line, ahead of one or more member names.

B Browse the member
C Copy the member
D Delete the member
E Edit the member
G Reset the member
J Submit the member
M Move the member
P Print the member
R Rename the member
T TSO command
V View the member.
W WS command

Note:

Member lists displayed with the M line command have a nine-character line command field to accommodate TSO commands, CLISTs, and REXX EXECs. See the **M-Display Member List** and **TSO Commands, CLISTs, and REXX EXECs** sections of the ***Data Set List Utility (Option 3.4)*** chapter of the *ISPF User's Guide Volume II* for more information.

When you use the R line command, enter the new member name in the **Prompt** field to the right of the member name.

When you press Enter, each member preceded by a line command is processed unless:

- The V (view), B (browse), or E (edit) line command is followed by another line command. When you return to the member list after Viewing, Browsing, or Editing a member, you must press Enter again to call any remaining line commands.
- You enter a line command for a member that was deleted. The names of deleted members are not removed from the member list until it is updated. Remove the line command that precedes the deleted member, and press Enter again. See "Updating a Member List" on page 149 for more information.
- You enter an R (rename) line command, but do not put a new name in the **Prompt** field. Enter a new member name, and press Enter again.

You can then do one of the following:

- Enter additional primary or line commands
- Scroll, if necessary, to bring additional members into view
- Enter the END command to return to the previous panel.

Figure 46 and Figure 47 show before and after examples that print members TEST and TEST1, delete member TEST8, and rename member TEST4 to OLDTEST.

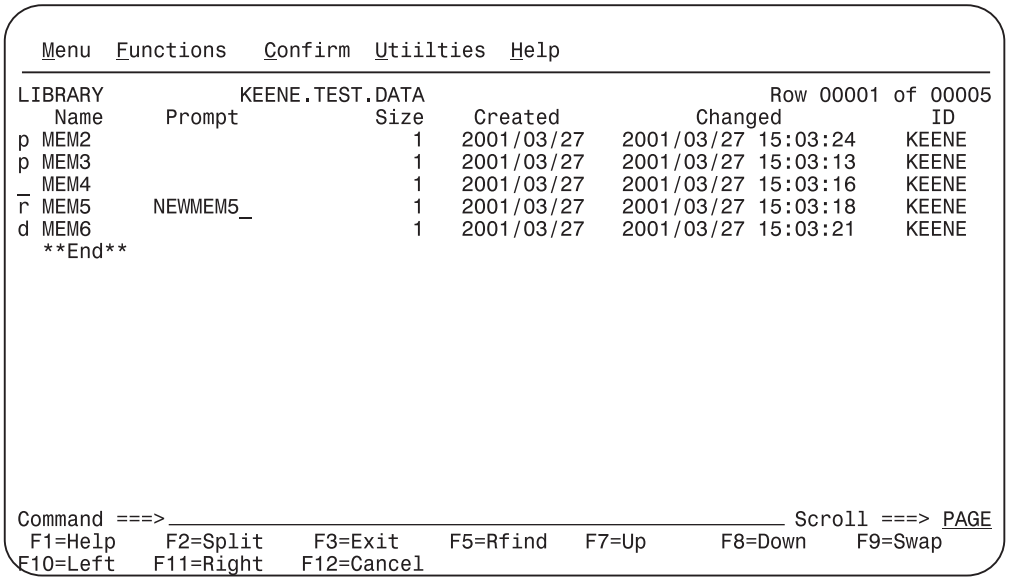


Figure 46. Library Utility before Print, Rename, and Delete (ISRUDMM)

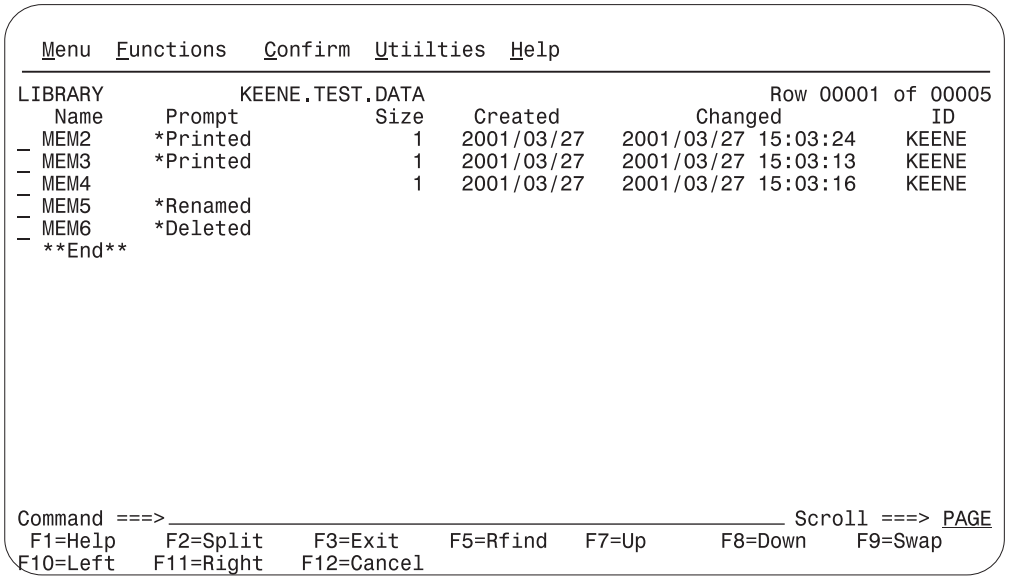


Figure 47. Library Utility after Print, Rename, and Delete (ISRUDMM)

Updating a Member List

When the member list is displayed again following completion of a function or command, it does not include:

- For View, Browse, and Edit:
  - New members created by recursive Edit calls
  - New members created using the CREATE command in EDIT do not display in an Enhanced Member List display, but do display when using the traditional member list processing.
  - Members created by another user
  - In split-screen mode, members created on another logical screen.

- For Library, Move/Copy, Data Set List, and Convert:
  - New names of members that have been renamed
  - Fewer member names when members are deleted (Library and Data Set List) or moved (Move/Copy).

To display an up-to-date list, return to the previous panel, leave the member name blank or enter a pattern. See "Displaying Member Lists" on page 135 for more information about displaying member lists.

To set your system for displaying a traditional member list when using the E,V, or B commands (Edit, View, Browse) specifically, see the **DSLIST Settings** section of Option 3.4 in the *ISPF User's Guide Volume II*.

Member List Positioning

After selected members have been processed, the member list is redisplayed and positioned based on the members selected.

- If a single member is selected, the member list is redisplayed with the selected member scrolled to the top of the display. However, if the CANCEL command is used to exit the selected member, the member list is not scrolled.
- If multiple members are selected and some selections are contained in the last screen displayed before the Enter key was pressed, the first selected member on that screen is scrolled to the top of the display.
- If multiple members are selected and none of the selections are contained in the last screen displayed before the Enter key was pressed, the first selected member in the member list is scrolled to the top of the display.

Data Set Passwords

A **Data Set Password** field is included on library and data set entry panels:

```
Data Set Password  . .          (If password protected)
```

The **Data Set Password** field contains the password for OS password-protected data sets. By assigning more than one password to the same data set, you can give some users read-only access while giving others read/write access.

Nondisplay input fields are used so that the passwords do not appear on the screen. When you specify a concatenated sequence of libraries, the password applies to all data sets in the sequence.

If you replace a long password with a shorter password, blank out the remaining spaces of the **Data Set Password** field.

You can use ISPF with either the TSO/VS2 Programming Control Facility (PCF) or the Resource Access Control Facility* (RACF*). PCF and RACF provide extensive facilities for data set security. However, when using either PCF or RACF, do not enter a password on the ISPF panels, because both facilities rely on your TSO user ID and logon password to identify you and check for proper authorization.

Format Definitions

A **Format Name** field is included on the View Entry Panel and on the Edit Entry Panel:

```
Format Name  . . . _______
```

The **Format Name** field contains the name of a format definition, which is used to view, browse, or edit a formatted data set. A *formatted* data set contains records that consist of subfields. The locations and lengths of these subfields are fixed throughout the data set. The formatted data set support in View, Browse, and Edit is particularly useful for data that contains double-byte character (DBCS) data but does not contain shift-out (SO) and shift-in (SI) characters.

The format name can consist of up to eight alphanumeric characters; the first one must be alphabetic.

A format definition can include Extended Binary Coded Decimal Interchange Code (EBCDIC) fields, DBCS fields, and mixed fields. If the specified format includes a mixed field definition, the **Mixed Mode** field is ignored, even if you select it. See "Mixed Mode" for information. For information about defining formats for formatted data sets, see the **Format Specifications Utility (Option 3.11)** section of the *ISPF User's Guide Volume II*. The Format Specifications utility is provided to support the IBM 5550 terminal that uses DBCS.

When formatted data is displayed, an attribute character that does not reside in the data set and is not stored in the data set precedes each field. Therefore, the column position on the display is different from the column position in the data set.

The allowable maximum length is decreased two bytes per field definition from the standard View, Browse, and Edit allowable maximum length.

Mixed Mode

A **Mixed Mode** field is included on the View Entry Panel and on the Edit Entry Panel:

```
_ Mixed Mode
```

The **Mixed Mode** field specifies whether or not you want to view, browse, or edit unformatted mixed data that contains both EBCDIC (single-byte) and DBCS (double-byte) characters. Use a slash to select mixed mode. If your terminal does not support DBCS, the value in this field is ignored.

DBCS strings are enclosed with SO (`X'0E'`) and SI (`X'0F'`) characters in unformatted mixed data. The SO character precedes the DBCS character string and the SI character follows the string.

If the view, browse, or edit line contains mixed data that are not valid, ISPF assumes the line can contain only EBCDIC characters. Examples of mixed data that are not valid include:

- Unpaired SO and SI characters
- Incorrect DBCS characters between SO and SI characters
- An odd number of bytes between SO and SI characters.

If you call View, Browse, or Edit from the Library utility (option 3.1) or the Data Set List utility (option 3.4), ISPF assumes that you want to use mixed mode.

If you want to view, browse, or edit DBCS data as EBCDIC data, you must do so in non-mixed mode. You can do this by operating from a terminal that does not support DBCS or by deselecting the **Mixed Mode** field.

Mixed Mode

In non-mixed mode, SO and SI characters are not treated as special characters; instead, they are treated as characters that cannot be displayed. Thus, you can view, browse, or edit the data in the conventional way.

You can also view, browse, or edit DBCS data in hexadecimal format, just as you would EBCDIC data. For information about specifying hexadecimal display, see the **HEX-Displaying Data in Hexadecimal Format** section of the ***View (Option 1)*** chapter of the *ISPF User's Guide Volume II*. You should not edit a record in hexadecimal format, however, when a DBCS string encroaches on the display boundary.

DBCS data that is not valid is not supported. If DBCS fields or DBCS strings in a mixed field contain any bytes with hexadecimal code ranging from `X'00'` to `X'3F'`, you may get unwanted results.

Partitioned Data Set Extended (PDSE)

The partitioned data set extended (PDSE) is a storage management subsystem formatted data set. Externally, the PDSE is very similar to a PDS. Internally, the PDSE contains a new directory structure, member format, and record format. A PDSE is indistinguishable from a PDS through most interfaces used to access a PDS directory or member. All ISPF functions support the PDSE.

You can concatenate a PDSE library with a PDS library if they have consistent record formats and logical record lengths. All functions in the Library Utility (option 3.1) support PDSEs with the exception of the compress function.

Packed Data Sets

The *packed* data set format allows you to use direct access storage devices (DASD) more efficiently. In this format, ISPF replaces any repeating characters with a sequence showing how many times the character is repeated. Before you can properly use data stored in this format as input to processing programs, such as compilers, you must first tell ISPF to unpack and expand the data.

The two requirements for using packed data sets are:

- To store data in packed format:
  - Enter the PACK ON Edit primary command while editing a data set or PDS member.
  - Select the **Pack Option** field (under To Data Set Options:) when copying or moving members using the Move/Copy utility (option 3.3).
- To unpack and expand packed data for processing, select the **Source Data Packed** field on the Foreground Selection panel or the Batch Selection panel. You must select this field if any of the input data, including that referred to in COPY or INCLUDE statements, is in packed format.

List and Log Data Sets

ISPF helps you get hardcopy listings of source modules, and it also maintains a log of significant user activities. These items are kept in data sets called the list data set and the log data set, respectively.

When needed, the two data sets are allocated automatically. They are temporary data sets named:

```
prefix.userid.SPFn.LIST
prefix.userid.SPFLOGn.LIST
```

prefix The data set prefix in your TSO profile. Used only if it is different from your user ID.

userid Your user ID.

n A number from 0 to 9.

If you have specified in your TSO profile a data set prefix that differs from your user ID, the data set names begin with your data set prefix, followed by your user ID. Once generated, these data sets remain open throughout your ISPF session. However, even though they are open, you can still process them by using the ISPF LIST and LOG commands.

List Data Set

The list data set is used for temporary storage for data to be printed at a later time. This data includes, for example, data written as a result of:

- Using the LIST service
- Issuing the PRINT, PRINT-HI, PRINTL, or PRINTLHI commands (but not PRINTG)
- Using Option 3 utilities.

To avoid generating an ISPF list data set, do not request any print functions.

Log Data Set

The log data set is used to capture data that can be useful for such things as diagnosing problems. This data includes, for example, data written as a result of:

- Using the LOG service
- Test and trace data such as:
  - ISPF TRACE mode data
  - Dialog Test option 7.7 dialog trace data.

Use the Log/List pull-down from the ISPF Settings panel action bar to prevent generating the ISPF log data set. However, if you use the Dialog Test option (7), allow generating the log data set because Dialog Test writes trace data to the log when you request it. Also, if Dialog Test finds an unexpected condition, problem data and error messages are written to the log.

Processing the Log and List Data Sets

You can process the log and list data sets either:

- During an ISPF session, using the LOG and LIST commands
- At the end of a session.

ISPF processes (prints, keeps, deletes) only data sets that it has allocated. Any attempt to process a log or list data set that has been preallocated by the user results in an appropriate ISPF message. Any references to ISPF processing of log or list data sets refer to data sets that ISPF has allocated. Users can supply routines to process preallocated data sets after ISPF has terminated.

How to Specify Log and List Data Set Processing Options

The log and list data set processing options can be specified through any of the following:

- Use of the LOG and LIST commands during an ISPF session.
- Use of the Log/List pull-down on the ISPF Settings panel for setting default options.

- The ISPF termination panel, which can display when you exit from ISPF. See "Log and List Data Set Processing at the End of a Session" on page 156 to find out under what conditions ISPF will display this panel.

Processing the Log and List Data Sets During an ISPF Session

The LOG and LIST commands allow you to process the log and list data sets, respectively, at any time during an ISPF session. The log and list data sets must have been allocated. You control the data set processing by specifying on the LOG or LIST command one of the three keyword options: PRINT, DELETE, or KEEP.

If you issue the LOG or LIST command with no parameter specified, ISPF displays a panel that allows you to select the data set processing options. The panels for the LOG and LIST commands are shown in Figure 48 and Figure 49 on page 155, respectively.

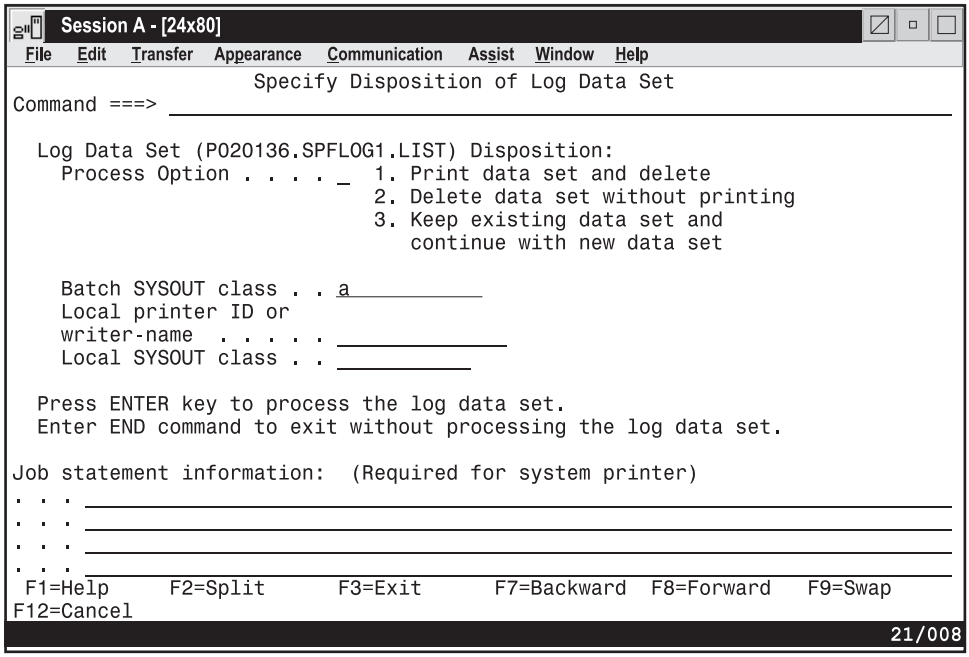


Figure 48. Log Data Set Defaults Panel (ISPLLP01)

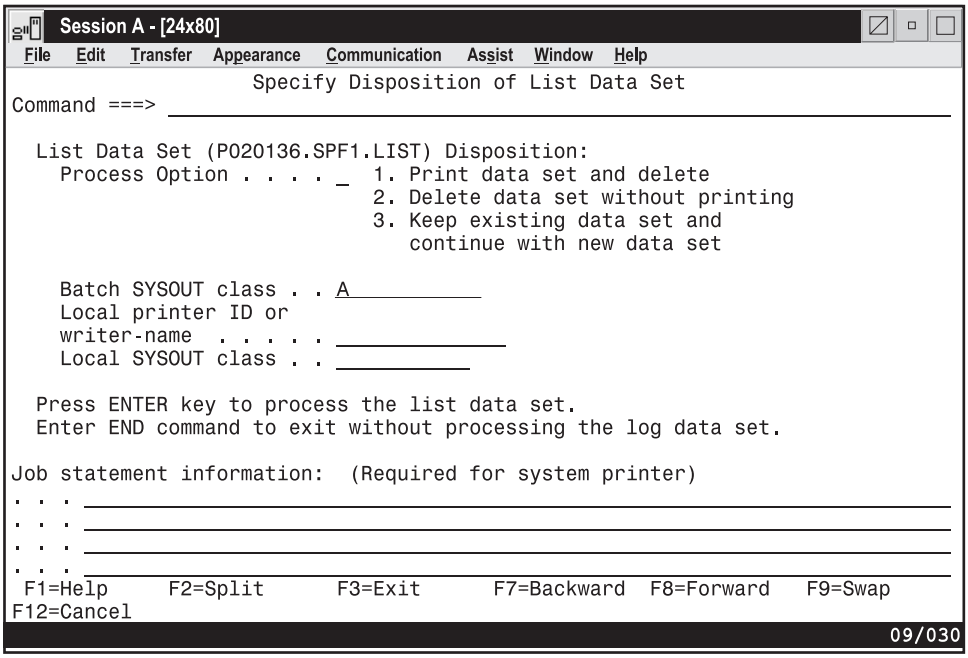


Figure 49. List Data Set Defaults Panel (ISPLLP02)

With the appropriate panel displayed, type in the process option of your choice. If you specify Print data set and delete, you must also specify a Batch SYSOUT class, or local printer ID or writer name. After you have typed in all information that you wish to specify, press Enter to pass the input to ISPF. ISPF takes the specified action for the data set and then returns you to the panel from which you issued the LOG or LIST command. ISPF issues a message indicating whether the action requested was successful.

If you issue the END command from the Log or List Data Set Defaults panel, ISPF returns you to the panel from which you issued the LOG or LIST command without processing the data set.

ISPF initializes the Log or List Data Set Defaults panel fields with the default values specified with the Log/List pull-down on the ISPF Settings panel. If a default disposition of Keep data set has been specified, ISPF translates the value to Keep data set and allocate new data set before displaying the panel. If you modify the process option field, the new value is used to process the data set; however, it is not saved in the system profile. All other fields modified on the panel are saved in the system profile and become the default values the next time the data set is processed.

If you issue the LOG or LIST command with the PRINT, DELETE, or KEEP option, ISPF does not display a panel. Specifying PRINT, DELETE, or KEEP on the command causes data set processing equivalent to specifying Print data set and delete, Delete data set, and Keep data set and allocate new data set, respectively, on the Log or List Data Set Defaults panel.

Two system variables, ZLOGNAME and ZLSTNAME, contain the fully qualified names of the log and list data sets, respectively. If either data set is not allocated or has not been used in the session, the corresponding system variable value is blank.

Note: The values of ZLOGNAME and ZLSTNAME are set to blank immediately after the log and list data sets have been processed because the data sets are freed by the LOG/LIST command processing. A new data set will not be allocated until it is written to. If you intend to use the log or list data set name for your processing, be sure to retrieve it before issuing the LOG or LIST command.

The system variables are summarized in *ISPF Dialog Developer's Guide and Reference*

Conditions for Using the LOG and LIST Commands: You can issue the LOG or LIST command from any command line except when any of the following applies:

- The command panel for the related log or list data set is active in any logical screen.
- The ISPF termination panel is active.
- The data set to be processed is not allocated or was preallocated.
- Dialog Test option 7.5 (Browse ISPF log) is active, and you are attempting to process the log data set.

An attempt to issue the LOG or LIST command in violation of any of these conditions results in ISPF issuing an appropriate message.

Log and List Data Set Processing at the End of a Session

Figure 50 shows the panel that ISPF displays at the end of a session if one of the following is true:

- The initial dialog began with the display of a menu, and the dialog is ended with the END command issued from that menu.
- The initial dialog began with the performance of a function, and the function ends with a return code of 0.
- The log and list data set processing defaults have not been specified, or the default values are not valid.

If the application ends with a non-zero return code, the termination panel is not displayed.

If the termination panel does not display for one of these reasons, the log and list data sets are processed using the default options.

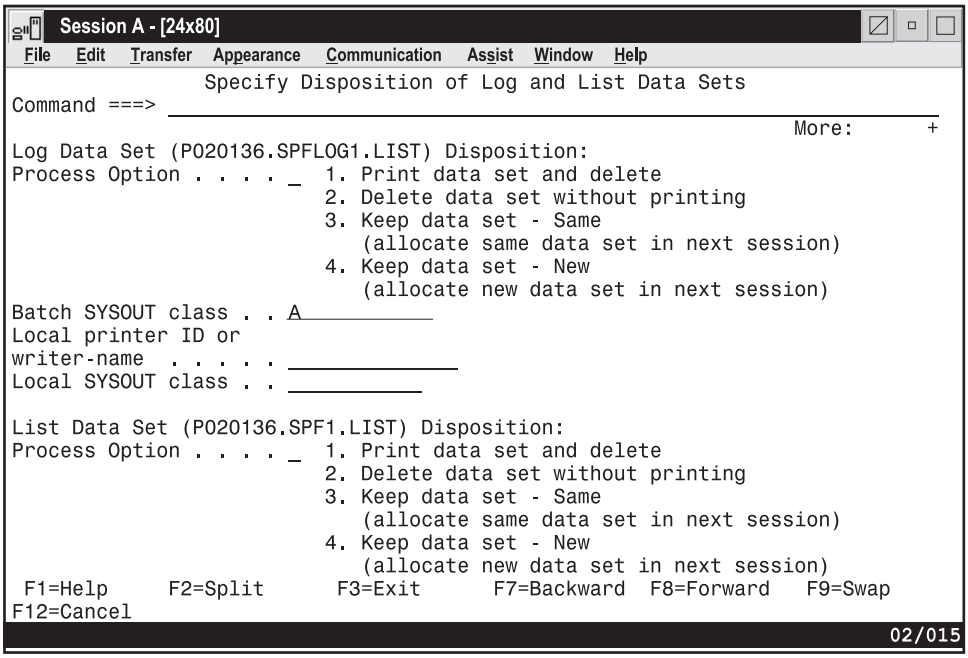


Figure 50. Specify Disposition of Log and List Data Sets Panel (ISPPFT03)

The valid process options shown in Figure 50 are described below.

Data Set Processing Options

For each term defined below, the first value shown is the processing option that you can specify on the Log/List pull-down from the ISPF Settings panel, on the Specify Disposition of Log and List Data Sets panel, or on the Log or List Data Set Defaults panel. The value in parentheses is the corresponding LOG or LIST command parameter.

1. Print data set and delete (PRINT)

Print the data set, then delete it. You must specify a Batch SYSOUT class or local printer ID or writer name.

- If the Batch SYSOUT class is specified, ISPF submits a background job to print and deletes the data set or sets.
- If a local printer ID or writer name is specified, ISPF uses the TSO PRINTDS command to route the data set to the specified printer or external writer program and then deletes the data set.

  Note: If you have selected Edit PRINTDS Command on the ISPF Settings panel (option 0), ISPF displays the Local Print Command Edit panel to allow you to intercept and edit the PRINTDS command before it processes. See “Editing the PRINTDS Command” on page 158 for additional information.

ISPF uses file tailoring services to print data on a system printer. Therefore, if this option is specified during an ISPF session, along with a Batch SYSOUT class, file tailoring must not be active on the logical screen from which the LOG or LIST command is issued. If an FTOPEN or FTINCL has been issued without a subsequent FTCLOSE, ISPF issues an appropriate message.

2. Delete data set without printing (DELETE)

Delete the data set.

3. Keep data set - Same
Not applicable to LOG or LIST command. Close and free the data set. Allocate the same data set the next session.

4. Keep data set - New (KEEP)
Close and free the data set. Allocate a different data set for the next session or the next time log or list information is generated in this session.

Editing the PRINTDS Command: If you have selected Edit PRINTDS Command on the ISPF Settings panel (option 0) and you specify a local printer ID or writer name on either the Log and List Data Set Termination Options panel or the Hardcopy Utility panel, ISPF displays the Local Print Command Edit panel shown in Figure 51 to allow you to edit the PRINTDS command before it processes.

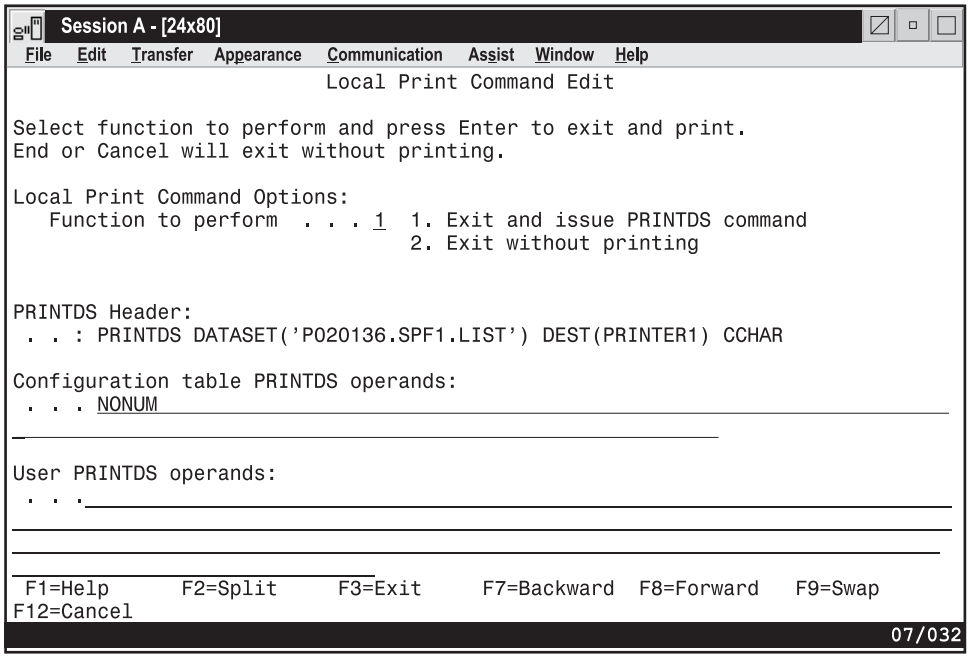


Figure 51. Local Print Command Edit Panel (ISPCHPLP)

The fields on this panel function as follows:

Function to perform
Specify the print function you want ISPF to perform:

1 Exit ISPF and issue the PRINTDS command (as edited below)

2 Exit ISPF without printing.

Note: If you arrive at this panel from ISPF termination processing, you will continue with termination and exit the product after your print request is issued or cancelled.

PRINTDS Header
This field cannot be edited. It contains the PRINTDS command, the data set name, the printer ID or writer name, and the CCHAR operand, if appropriate.

Configuration table PRINTDS operands
These operands operate at a system level and can be altered only in the ISPF Configuration table.

User PRINTDS operands
Enter additional operands (for example, COPIES or FORMS). These operands can be edited and are saved in the application command table.

If you enter CANCEL (or select Cancel), the PRINTDS command is not issued. If you enter END or RETURN or use a jump function, the PRINTDS command is issued and you receive a completion message.

Foreground and Batch Output Listings

The following additional listing data sets are allocated as needed for foreground or batch processing:

```
prefix.userid.list-id.LIST
prefix.userid.list-id.LINKLIST
prefix.userid.list-id.TERM
prefix.userid.list-id.TESTLIST
```

prefix The data set prefix in your TSO profile. Use it only if you have one and it is different from your user ID.

userid Your user ID.

list-id The name specified in the **List ID** field on the foreground or batch data entry panel. This name is required for sequential data sets. However, for partitioned data sets, the member name becomes the default *list-id* if the List ID field is blank.

The particular data set names you use depend on the foreground or batch processing option chosen.

For batch processing, the output can either be directed to a list data set or printed as part of the batch job. When batch processing is finished, you can browse the list data set, and then use the Hardcopy utility (option 3.6) to print it. Using this utility, show whether you want to keep the data set or delete it after printing. ISPF does not delete these data sets when you end ISPF.

For the foreground option, the output listing is directed to a list data set and automatically displayed for browsing. When you end the browse function, ISPF displays a selection panel that allows you to choose whether to print, keep, or delete the list data set. Again, ISPF does not delete this data set when you end ISPF.

Other Temporary Data Sets

If you are using virtual I/O (VIO), you can allocate space for temporary data sets, and then VIO assigns them system-generated names. Otherwise, ISPF allocates temporary control and listing data sets, as needed, for its own internal use. You are usually not aware of their existence. They are assigned the following names:

```
prefix.user-id.SPFTEMPn.CNTL
prefix.user-id.SPFTEMPn.LIST
prefix.user-id.appl-idzzzz.BACKUP
prefix.user-id.appl-idzzzz.BACKUPI
prefix.user-id.SPFxxx.OUTLIST
```

prefix The data set prefix in your TSO profile. It is used only if you have one and it is different from your user ID.

Other Temporary Data Sets

userid Your user ID.

n A number from 1–4, depending on which logical screen is active.

appl-id
The application ID.

zzzz A number from 0001–0008, or higher if customized, controlled by the edit recovery table (*appl-id* EDRT for the EDREC service and *appl-id* EIRT for the EDIREC service) and the number of concurrent edit calls that are active.

xxx A number generated by ISPF, which has a range of 100–999.

These data sets are deleted:
- By edit recovery when the data sets are no longer needed
- When you specifically request that they be deleted.
- By Move/Copy when no IEBCOPY errors are encountered.

Job Statement Information

ISPF allows you to submit Batch jobs for printing and language processing. However, before submitting a Batch job, you must supply job statement information. For this purpose, four lines are provided on each job submission panel.

You can use the lines that contain //*:
- As continuation lines by removing the asterisk (*)
- To enter other JCL statements, such as JOBLIB DD.

If you do not need these lines, you can blank them out. Blank lines are not submitted to the job stream.

Running a Sample ISPF Session

This section provides an example of an ISPF session. For new users, it is a quick introduction to ISPF. For users with previous ISPF experience, it is a quick review. It can also be useful as a demonstration that ISPF has been properly installed and is operational.

The scenario requires the installation of a data set named SYS1.SAMPLIB. This data set is included on the ISPF basic distribution tape and should contain the following four members:

ISRASM Sample assembler source
ISRCOBOL Sample COBOL source
ISRFORT Sample FORTRAN source
ISRPLI Sample PL/I source.

During this scenario, member ISRASM is copied from SYS1.SAMPLIB to a user data set. The other three members are not used.

For this scenario, the TSO Assembler Prompter, program number 5734-CP2, is installed and available. If the Assembler Prompter is not available at your installation, skip steps 35 through 48.

Action	Result
1. Log on to TSO.	A `Ready` message appears on your screen.
2. Type `ISPF` (or the appropriate alias) and press Enter.	The ISPF Primary Option Menu appears.

Action	Result
3. On the Option line, type 3 to select the Utilities option. Then press Enter.	The Utility Selection Panel appears.
4. On the Option line, type 2 to select the Data Set utility. Then press Enter.	The Data Set Utility panel appears.

5. On the Option line, type A to allocate a new data set. Specify an ISPF library by typing the following information, but substitute your first name in the Group field:
`Project . . your_user_id`
`Group . . . name`
`Type . . . . ASM`
 	
Now press Enter.
	The Allocate New Data Set panel appears. Note: The Project name must be defined as a valid high-level identifier on your installation. Generally, user-ids are defined as such. If this is not true for your installation, contact your system programmer for information about what project names are valid for your system.

6. Type the following information:
Note: Leave ALL fields blank except the following.
`Space units . . . . . TRKS`
`Primary quantity. . . . 2`
`Secondary quantity. . . 1`
`Directory blocks. . . . 1`
`Record format . . . . . FB`
`Record length . . . . . 80`
`Block size . . . . . 3120`
 	
Now press Enter.
	Data set your_user_id.*name*.ASM is allocated on scratch volume. The Data Set Utility panel reappears.

7. Select option A again and specify the following ISPF library, again substituting your first name in the Group field:
`Project . . your_user_id`
`Group . . . name`
`Type . . . . OBJ`
 	
Press Enter.
	The Allocate New Data Set panel reappears.

8. Leave everything the same. Just press Enter.	Data set your_user_id.*name*.OBJ is allocated. The Data Set Utility panel reappears.
9. Press F3.	The Utility Selection Panel reappears.
10. Press F3 again.	The ISPF Primary Option Menu reappears.
11. Type 3.3 to select the Move/Copy utility, bypassing the Utility Selection Menu. Press Enter.	The Move/Copy Utility panel appears.
12. Now you will copy a data set. On the Option line, type C to select `Copy data set or member(s)`. Then, under From Other Partitioned or Sequential Data Set:, enter the following data set name: `Data Set Name . . . 'SYS1.SAMPLIB(ISRASM)'` Press Enter.	A panel titled COPY FROM SYS1.SAMPLIB(ISRASM) is displayed.

13. Under To ISPF Library:, type the following values:
`Project . . your_user_id`
`Group . . . name`
`Type . . . . ASM`
 	
Press Enter.
	Member ISRASM is copied from data set SYS1.SAMPLIB to ISPF library your_user_id.*name*.ASM. Then, the Move/Copy Utility panel reappears.

14. Press F3.	The ISPF Primary Option Menu appears.
15. Now you will browse member ISRASM. On the Option line, type 1 to select View and press Enter.	The View Entry Panel appears. Select Browse Mode.

Running a Sample ISPF Session

Action	Result

16. Type the following under ISPF LIBRARY:
`Project . . your_user_id`
`Group . . . name`
`Type . . . . ASM`
 	
Note: Leave the Member field blank.
 	
Press Enter.
	A member list is displayed, showing ISRASM as the only member in the your_user_id.*name*.ASM library.

17. Move the cursor to the left of ISRASM. Then type `S` to select ISRASM and press Enter.	A panel titled BROWSE your_user_id.*name*.ASM (ISRASM) appears. This is the first page of member ISRASM.
18. Press F8 to scroll ahead one page.	The second page of ISRASM appears.
19. Press F7 to scroll backward one page.	The first page of ISRASM reappears.
20. Type `FIND COMMENT` on the Command line and press Enter.	The cursor moves to the first occurrence of the character string `COMMENT` and the string is intensified. Also, the message `CHARS 'COMMENT' FOUND` is displayed in the upper-right corner of the screen.
21. To find the next occurrence of `COMMENT`, press F5, the RFIND command.	The cursor moves to the second occurrence of `COMMENT` and once again the string is intensified.
22. Press F3.	The member list reappears.
23. Press F3 again.	The View Entry Panel reappears.
24. Press F3 one more time.	The ISPF Primary Option Menu appears.
25. Now you will edit member ISRASM. On the Option line, type `2` to select Edit and press Enter.	The Edit Entry Panel appears.
26. Type `ISRASM` in the Member field and press Enter.	A panel titled EDIT your_user_id.*name*.ASM(ISRASM) appears. This is the first page of member ISRASM.
27. On the Command line, type `FIND COMMENT` and press Enter to find the line containing the character string `COMMENT`.	The cursor moves to the first occurrence of the character string and the line number is intensified.
28. Delete `COMMENT` by pressing the Erase EOF key.	`COMMENT` is erased. Any characters to the right of `COMMENT` are erased, also.
29. Press F5 to find the next occurrence of `COMMENT`.	The cursor moves to the second occurrence of `COMMENT` and the line number is intensified.
30. Move the cursor to the sequence number of the line below `COMMENT`, then move the cursor up one line. Repeat the `COMMENT` line by typing `R` over the first digit of the line number and pressing Enter.	The line is repeated.
31. On the Command line, type `AUTOLIST ON`, and press Enter.	Sets autolist mode on for automatic source listings.
32. Try out more Edit commands if you like, but remember: this program will be assembled later.	HAVE FUN!
33. Press F3.	Member ISRASM is saved in data set your_user_id.*name*.ASM and a listing of the member is placed in the ISPF list data set. The Edit Entry Panel appears.
34. Press F3 again.	The ISPF Primary Option Menu appears.
35. Now you will assemble the program. On the Option line, type `4` and press Enter to select the Foreground option.	The Foreground Selection Panel appears.

Action	Result
36. On the Option line, type 1 and press Enter to select Assembler.	The Foreground Assembler panel appears.

37. Type the following values:
`Project . . .your_user_id`
`Group . . . .name`
`Type  . . . .ASM`
`Member  . . .ISRASM`
`List ID . . .ASMTEST1`
`Assembler Options:`
`  ===> TEST,RENT`
Press Enter.
	The Assembler Prompter is called. Terminal output is written at the bottom of a blank screen. When *** is displayed, press Enter to continue.
	Note: The assembly may take a few minutes.
	The assembly listing is displayed in Browse mode.

38. Browse the listing using the scroll commands. Then press F3.	The Foreground Print Options panel appears.
39. On the Option line, type K to select Keep data set (without printing) and press Enter.	The assembled program is saved in the list data set and the Foreground Assembler panel reappears.
40. Press F3.	The Foreground Selection Panel reappears.
41. Press F3 again.	The ISPF Primary Option Menu appears.
42. On the Option line, type 5 to select the Batch option and press Enter.	The Batch Selection Panel appears.
43. On the Option line, type 1 to select Assembler, enter job statement information as required by your installation, and press Enter.	The Batch Assembler panel appears.

44. Type the following values:
`Project . . . your_user_id`
`Group . . . . name`
`Type  . . . . ASM`
`Member  . . . ISRASM`
`List ID  . . . . . ASMTEST2`
`Assembler options:`
`  Term  . . . TERM    (TERM or NOTERM)`
`  Other . . . OBJ,NODECK,LIST`
Press Enter.
	JCL for assembling ISRASM is generated. The Batch Selection Panel reappears. The message "Job step generated" appears in the upper right corner of the panel.

45. Press F3.	The job is submitted. The job name is displayed at the bottom of the screen. When *** is displayed, press Enter. The ISPF Primary Option Menu appears.
46. On the Option line, type 6 to select the Command option. Press Enter.	The ISPF Command Shell panel appears.
47. Type the following command: `===> STATUS` Press Enter.	The status of your job appears. When *** is displayed, press Enter. The ISPF Command Shell panel is displayed again with the previous command still showing.
48. Press F3.	The ISPF Primary Option Menu reappears.
49. Move the cursor to **Help** on the action bar. Press Enter. On the resulting pop-up window, select **18**.	The beginning of the ISPF tutorial appears. Follow the directions to learn more about ISPF. When you have finished, press F3 to return to the ISPF Primary Option Menu.

If you have followed this scenario, you should have an Assembler listing in a data set named either *prefix*.*userid*.ASMTEST1.LIST or *userid*.ASMTEST1.LIST. Also,

when the job that you submitted has finished, you will have another listing in a data set named either *prefix*.*userid*.ASMTEST2.LIST or *userid*.ASMTEST2.LIST. Try the Browse option to review the listings and try experimenting with other ISPF options. When you have finished, return to the ISPF Primary Option Menu.

Action	Result
50. With the ISPF Primary Option Menu on the screen, press F3.	The ISPF Specify Disposition of Log and List Data Sets panel appears.
51. Select the process option to print and delete both the log and list data sets, and fill in the job statement information as required by your installation. Press Enter.	Data sets will be printed, then deleted through batch jobs submitted by ISPF. The job name is displayed at the bottom of the screen.
52. You are now out of ISPF. To leave TSO, type `LOGOFF` and press Enter.	This is the end of this usage scenario.

Chapter 6. Getting Ready to Run on MVS

This chapter helps you prepare to use ISPF data-element libraries. ISPF data elements include such things as panels and messages. Libraries to be accessed during processing of a dialog must be defined to ISPF. This chapter describes the kinds of data-element libraries required when ISPF is active. It also describes and provides examples of how to define library data sets to ISPF, both prior to starting a session, and dynamically during a session.

Setting Up ISPF Libraries

To set up libraries for developing and testing dialogs, follow these steps:

1. Set up the panel, message, skeleton, table, and program libraries for the application. Allocate new partitioned data sets.
2. Create a CLIST or REXX command procedure that contains the necessary ALLOCATE statements to allocate the libraries. Concatenate the application libraries ahead of the libraries required by ISPF, as previously described in "Library Concatenation" on page 131.

   Note: You can use the LIBDEF service to dynamically allocate libraries instead of allocating them prior to invoking ISPF. For further information see the *ISPF Dialog Developer's Guide and Reference*
3. Create the panels, messages, and skeletons by editing directly into the application libraries.
4. Create the dialog functions and ensure that the load modules are in libraries accessible to ISPF.

   Functions coded as program modules *must* be link edited. When a function is link edited, the ISPLINK subroutine must be included (explicitly or by automatic call) in the load module. ISPLINK is distributed in load module format and can be placed in a system library for automatic call during link edit.
5. Invoke the application. To do this, add an ISPSTART command to the command procedure created in step 2. The ISPSTART command should start the application using the appropriate PANEL, CMD, or PGM parameter. Users can start the application by using this command procedure or by selecting the application from the master menu or another menu.

Allocating Required ISPF Libraries

The libraries described in Table 8 are partitioned data sets required for operation of ISPF in the MVS/TSO environment:

Table 8. Required Partitioned Data Sets

DDNAME	Description	RECFM	LRECL	BLKSIZE
ISPPLIB	Panel Library	FB	80	3120
ISPMLIB	Message Library	FB	80	3120
ISPSLIB	Skeleton Library	FB	80	3120
ISPTLIB	Table Input Library	FB	80	3120
ISPPROF	User Profile Library	FB	80	(see note)
SYSPROC	Command Procedures Lib	FB	80	3120

Note: The block size can be established by the application. It must be a multiple of 80.

The panel, message, skeleton, and table input libraries are distributed with ISPF. As distributed, the libraries have the characteristics listed above. These libraries can be reblocked by the installation to a larger block size. In addition, the panel, message, and skeleton libraries can be copied into a variable record format. The maximum length records supported are 160 for panels, 80 for messages, and 255 for skeletons. If data sets having unequal record lengths are to be concatenated, the record format must be variable. If you have preprocessed any panels in the panel library, they must be reprocessed using the ISPPREP utility after changing the panel library's record size or record format. Preprocessed panels will not function correctly if copied directly to a data set with a different record size or format.

Table 9 contains the LRECL limits which are enforced during ISPF initialization:

Table 9. LRECL Limits during ISPF Initialization

DDNAME	Description	RECFM	Minimum LRECL	Maximum LRECL
ISPPLIB	Panel Library	FB VB	80 84	160 164
ISPMLIB	Message Library	FB VB	80 84	80 84
ISPSLIB	Skeleton Library	FB VB	80 84	255 259

The VB libraries require the LRECL to contain 4 extra bytes for the record descriptor word.

Note: Use of the BUFNO parameter on allocation of ISPF libraries is not supported.

Problems can occur when using file tailoring services in conjunction with other services (EDIT, COPY, ...) that result in modifying the data set members in the ISPSLIB concatenation. ISPSLIB is the input skeleton library, and it is assumed to be a static library. FTINCL obtains existing DCB/DEB information based on the last OPEN done against ISPSLIB by ISPF.

It is recommended that applications that use file tailoring and that also modify members of ISPSLIB use the LIBDEF service for ISPSLIB to point to the application's skeleton library. Additionally, the application should check for any changes to the data set information (DCB/DEB) prior to invoking file tailoring services. If there has been a change, then the application should issue a NULL LIBDEF for ISPSLIB and then re-issue the original LIBDEF for ISPSLIB. This forces the ISPSLIB library to close and then re-open.

ISPF assumes that ISPSLIB is a static library. When you make allocations, consider limiting the possibility of extents by allocating the skeleton with the largest optimal block size.

There is a separate profile library for each end user. Its contents are dynamically generated and updated while ISPF is running. There is also a unique profile library for each national language version.

The recommended data set names for these libraries are shown below. Check with your ISPF system administrator to determine if these are the actual data set names used at your installation.

DDNAME	**DSNAME**
ISPPLIB	ISP.SISPP*xxx*
ISPMLIB	ISP.SISPM*xxx*
ISPSLIB	ISP.SISPS*xxx* ISP.SISPSLIB
ISPTLIB	ISP.SISPT*xxx*
ISPPROF	User-selected. Unique for each national language used.
SYSPROC	ISP.SISPEXEC ISP.SISPCLIB

where *xxx* is the placeholder for the specific language you are using. For example:

xxx	**Language**
ENU	US English
DES	Swiss German
DEU	German
JPN	Japanese
ENP	Uppercase English.

You should concatenate application libraries for panels, messages, skeletons, and tables ahead of the corresponding ISPF libraries using the DDNAMEs shown above. The application libraries must have the same data set characteristics as the required libraries, as described above. For example, assume that application XYZ uses the following partitioned data sets for panels, messages, skeletons, and tables:

```
XYZ.PANELS
XYZ.MSGS
XYZ.SKELS
XYZ.TABLES
```

You would issue the following allocations:

```
//ISPPLIB  DD DSN=XYZ.PANELS,DISP=SHR
//         DD DSN=ISP.SISPPxxx,DISP=SHR

//ISPMLIB  DD DSN=XYZ.MSGS,DISP=SHR
//         DD DSN=ISP.SISPMxxx,DISP=SHR

//ISPSLIB  DD DSN=XYZ.SKELS,DISP=SHR
```

```
//         DD DSN=ISP.SISPSxxx,DISP=SHR
//         DD DSN=ISP.SISPSLIB,DISP=SHR

//ISPTLIB  DD DSN=XYZ.TABLES,DISP=SHR
//         DD DSN=ISP.SISPTENU,DISP=SHR

//ISPPROF  DD DSN=USERAA.ISPF.PROFILE,DISP=OLD

//SYSPROC  DD DSN=ISP.SISPEXEC,DISP=SHR
//         DD DSN=ISP.SISPCLIB,DISP=SHR
```

where *xxx* is the placeholder for the specific language you are using. For example:

xxx	**Language**
ENU	US English
DES	Swiss German
DEU	German
JPN	Japanese
ENP	Uppercase English.

These allocations must be performed before you start ISPF. They can be done in the user's TSO LOGON procedure by using DD statements, as shown above, or in a CLIST or REXX command procedure by using the corresponding TSO ALLOCATE commands.

Allocating Optional Table and File Tailoring ISPF Libraries

The data sets described in Table 10 are optional. You must allocate them only if an application uses table or file-tailoring services.

Table 10. Table and File-Tailoring Data Sets

DDNAME	Description	RECFM	LRECL	BLKSIZE
ISPTABL	Table Output Library	FB	80	(See note)
ISPFILE	File-Tailoring Output	FB/VB	255 max.	

Note: The block size can be established by the application. It must be a multiple of 80.

The table output library must be a partitioned data set. The ISPTABL DDNAME that defines the table *output* library can specify the same data set as the table *input* library, DDNAME ISPTLIB. The output and input data sets must be the same if the updated version of a table is to be reprocessed by the same dialog that updated it.

You must allocate the table output library to DDNAME ISPTABL prior to using table services. ISPF includes ENQ logic to prevent simultaneous updates. ISPTABL must *not* specify a concatenated sequence of data sets.

The dialog can dynamically allocate ISPTABL, and can free it upon completion of its use. ISPTABL should be allocated with DISP=SHR, even though it specifies an output data set. The TSO Programming Control Facility II (PCF) cannot protect the table output library from unauthorized updating if the library is allocated DISP=SHR. The library can be either protected by RAC or allocated with DISP=OLD and protected by PCF.

File-tailoring output can be written to a temporary sequential data set provided by ISPF. The temporary data set is allocated automatically, so there is no need for the dialog to allocate a data set. The fully qualified name of the temporary data set is available in system variable ZTEMPF. The DDNAME is available in ZTEMPN. This temporary data set always uses 80-character fixed-length records.

If the temporary data set is not used, file-tailoring output can be written to either a partitioned or a sequential data set. Both fixed-length and variable-length records are permitted. The maximum logical record length is 255 bytes. A data set must be allocated to DDNAME ISPFILE before starting file-tailoring services. The dialog can dynamically allocate the output library, and can free it upon completion. For a sequential data set, ISPFILE must be allocated with DISP=OLD. For a partitioned data set, it can be allocated with DISP=SHR but cannot be protected by the Program Control Facility II (PCF) unless it is allocated with DISP=OLD. ISPFILE must not specify a concatenated sequence of data sets.

Allocating Optional Image ISPF Library

The data set described in Table 11 is not a required ISPF library. You must allocate it only if an application in GUI mode uses images.

Table 11. Image Data Set

DDNAME	Description	RECFM	LRECL	BLKSIZE
ISPILIB	Image Input Library	FB	80	(See note)

Note: The block size can be established by the application. It must be a multiple of 80.

If you plan to use ISPF's image support, you must allocate the image input data set to ddname **ISPILIB** before using the images. You can accomplish this by allocating the image input data set before you invoke ISPF. Image files in the *Graphic Interchange Format* (GIF) should reside in this input data set. This data set must be a partitioned data set. It can be allocated with DISP=SHR.

Note: The LIBDEF service does *not* result in the allocation of ISPILIB.

ISPF ships sample image files in the sample library SISPSAMP. The ISPF panel ISR@PRIM uses three of the sample image files — ISPFGIFL,ISPFGIFS, and ISPEXIT. You can copy the sample image files to your own image input data set allocated to ddname ISPILIB.

Allocating CLIST, REXX, and Program Libraries

Dialog functions that are coded as CLIST or REXX command procedures can be in a procedure library that has been allocated to DDNAME SYSPROC prior to starting ISPF. A REXX command procedure can also be allocated to the SYSEXEC DDNAME. The SYSEXEC DDNAME is described in *TSO/E Version 2 REXX User's Guide*

You must link edit dialog functions that have been coded as programs. The load module can reside in a step library, a system link library (such as SYS1.LINKLIB), or the link pack area. Alternatively, it can be in a partitioned data set (RECFM=U) allocated to DDNAME ISPLLIB(DISP=SHR). This library (the ISPF Link Library)

can be used for testing new dialogs that contain program-coded functions. If used, it must be allocated prior to starting ISPF. ISPLLIB can specify a concatenated sequence of partitioned data sets.

ISPLLIB is used as a task library when fetching load modules. It is searched prior to the system link libraries and the link pack area. If ISPF product modules are kept in a step library and a task library (ISPLLIB) is used, the data sets containing the ISPF product modules should be included in the ISPLLIB concatenation sequence as well as the step library concatenation. If a program is to be used in split-screen mode it should be linked as reentrant or nonreusable.

The exception to the search order described previously is the SISPSASC library. Modules in it are not searched for using the ISPLLIB task library. SISPSASC must be in STEPLIB or LNKLST if you are using the ISPF C/S feature. For more information about SISPSASC, refer to *ISPF Planning and Customizing*

Allocating DBCS Libraries

DBCS users can use alternate message, panel, and skeleton libraries. To do so, the DBCS versions of the libraries must have been allocated using the DDNAMEs:

ISPMALT Alternate message library
ISPPALT Alternate panel library
ISPSALT Alternate skeleton library.

You can allocate these libraries when you allocate the distributed ISPF libraries. If the alternate libraries are allocated and the terminal has DBCS capability, ISPF uses the alternate libraries. If either of these two conditions is not satisfied, ISPF uses the distributed libraries.

Selecting the National Language for ISPF Sessions

An ISPF session can be run in any installation-supported national language. Before starting ISPF with the ISPSTART command, a user must perform the necessary allocations. For example, command procedure ISPFE might be issued for an English session or ISPFG for a German session.

The same set of DDNAMEs (ISPPLIB, ISPMLIB, ISPSLIB, ISPTLIB, and ISPPROF) must be allocated regardless of the command procedure used. At logon time, the necessary allocations for the national language at an installation can be performed by a CLIST or REXX logon procedure.

The language in which a session runs is reflected by the value (not always the full language name) in read-only system variable ZLANG, which is available to dialogs running under ISPF. The default value for session languages is specified when ISPF is installed, and is discussed in *ISPF Planning and Customizing* You can override the default session language with an alternate language keyword on the ISPSTART command. See the *ISPF Dialog Developer's Guide and Reference* for the exact syntax to use.

By specifying a default session language, the installation can ensure that both ISPF initialization messages and the normal session messages are in the default language. Even if you override the session default language with an alternate language using an ISPSTART language keyword, some of the first initialization messages, issued prior to the command scan, are in the default session language. However, any messages issued after processing of the ISPSTART parameters are in the language specified by the keyword.

If the terminal does not support DBCS, and the default language (or the alternate language selected by the language keyword) requires DBCS, ISPF uses English as the session language.

In cases where the session language requires DBCS, certain messages are always issued in English. These messages are:

- ISPF Main task abend.
- ISPF Subtask abend.
- The following required module for the selected language could not be loaded.
- ISPF command not allowed. You are already under ISPF.
- Invalid environment; TSO/E version 2.1 or later required.

Part 2. Appendixes

Appendix A. Installation Considerations for the Client/Server

This appendix provides information on the following topics that you need to be aware of before you install the ISPF Client/Server (ISPF C/S) Workstation Agent (ISPF WSA) component on your programmable workstation:

- Assumptions
- Required hardware and software
- Configuration parameters
- GUI-Mode CLIST or STARTUP procedure
- Configuration of commnunications.

See the **Download Data Set to Workstation Utility (Option 3.7)** section of the *ISPF User's Guide Volume II* for information on using option 3.7 to install the workstation software.

Assumptions

The ISPF C/S Component Install Utility makes the following assumptions about your operating environment:

Workstation-Host Connection

An APPC connection exists between your workstation and the host, and your workstation is running under OS/2. Or, a TCP/IP connection exists between your workstation and the host, and your workstation is running under one of the following supported operating systems:

- IBM OS/2
- Microsoft Windows NT/95/98/2000
- AIX
- HP-UX
- Solaris (SUN).

File Download Capabilities

The ISPF C/S Component Install Utility provides an automated download procedure for workstation implementations using TCP/IP, provided that the TCP/IP on your workstation supports the File Transfer Protocols (FTP). If a workstation connection has been established, then the ISPF C/S component can be downloaded using the workstation connection. If your workstation does not support FTP, or you do not have an established connection, it is assumed that you have a means of transferring the ISPF C/S workstation programs down from the host files to the workstation. Although the ISPF C/S Install Utility does not provide an automated download procedure for all users, it does provide information that can assist in the download process. See the **Download Data Set to Workstation Utility (Option 3.7)** section of the *ISPF User's Guide Volume II* for more information.

TCP/IP File Transfer Protocol

The ISPF C/S Install Utility uses TCP/IP's file transfer protocol (FTP). The utility assumes that FTP is operational at your workstation. An operational FTP requires that:

- FTP be configured and started on the workstation.
- The workstation FTP server (FTPD), or "Daemon," be active.

- The users have the authority to transfer files. Users are authorized through entries in the FTP Daemon access protection file. See "Configuration Parameters" for additional information.

Required Hardware and Software

For the hardware and software requirements for ISPF, refer to the *OS/390 Release 10 Planning for Installation* manual, document number GC28–1726–08.

Configuration Parameters

Before you run the ISPF C/S Install Utility, gather the following installation-unique parameters:

IP Address
The TCP/IP address, Internet address, workstation ID, or Internet address alias.

User ID
This is the FTP user name that must have write access to the directory where the workstation ISPF Workstation Agent code is to be installed.

Password
Password for the FTP User ID specified.

Directory
The workstation directory where you want the utility to install the programmable workstation programs. The User ID must have write access to this directory.

The user authorization is granted through an entry in the FTP Daemon access protection file. If you do not have an entry in this file, you will need to create one. Refer to the TCP/IP documentation for details on creating a user entry in the FTPD Access Protection file.

Appendix B. Configuring Communications for the ISPF Client/Server

Either a TCP/IP or an SNA APPC connection must be enabled to support ISPF Client/Server (C/S) communication between an MVS host and a workstation. The following sections describe what must be configured at the host and at a workstation for both TCP/IP and SNA APPC connections. Configuration of 3270 terminal type connections that support user logon to MVS TSO is not described here. A configuration fastpath is provided at the beginning of the discussion of each type of connection.

Although a few elements of ISPF C/S configuration must be coordinated between a workstation and the MVS host, ISPF users will generally be more concerned with workstation requirements. MVS system or network support personnel will generally be more concerned with MVS host requirements.

Note: All necessary TCP/IP or APPC communications software must be initialized fully before starting the ISPF Workstation Agent component. The WSA does not attempt to intialize the communications protocol again after it receives the first initialization error.

Configuring TCP/IP connections

The following table provides a fastpath for configuring TCP/IP communications to support ISPF C/S.

TCP/IP Requirements Fast Path

Workstation	MVS host
A numeric internet address must be defined for TCP/IP on the workstation (such as, 9.67.224.96).	The numeric internet address defined for TCP/IP on the workstation must be specified on the *ISPSTART* command or from the GUI settings panel to invoke ISPF WSA.
No workstation requirement matching this host requirement →	It must be possible for ISPF to locate the TCP/IP configuration data set containing the TCP/IP started task name. Keywords **SAS/C_TCPIP_DATA_VALUE** and **SAS/C_TCPIP_PREFIX_VALUE** in the **ISPF Configuration Table Keyword File** can be used to determine the appropriate data set.

TCP/IP Requirements Detail

An internet address is the 32-bit address assigned to a TCP/IP host and usually represented in dotted decimal form. For example:

```
9.67.224.96
```

The address identifies both a physical network within the internet and an individual host within that network.

A TCP/IP host can be an MVS system or an OS/2, Windows, or UNIX workstation. For the purpose of ISPF C/S communications you need to know the

dotted decimal internet address associated with the workstation on which the ISPF WSA graphical user interface is to be displayed. In many enterprises the workstation internet address is obtained from a network administrator and assigned to the workstation during installation and customization of the TCP/IP workstation software. For information on installation and customization of the TCP/IP software used on your workstation, consult the documentation supplied by your TCP/IP software vendor. If you have already installed TCP/IP on your workstation and you do not know your TCP/IP internet address, but you do know your TCP/IP workstation host name (such as JSMITH, ACCT23), you can determine the internet address as follows:

If you have this TCP/IP workstation software	**do this on the workstation**
• IBM TCP/IP for OS/2	Select the *Command Prompts* folder after selecting the *OS/2 System* icon. Select *OS/2 Window* from the folder and at the displayed OS/2 command prompt enter the command *host* <hostname>. The numeric internet address of the workstation host identified by <hostname> should appear in the reply.
• Windows TCP/IP	From an MS/DOS command prompt enter **winipcfg**. The numeric internet address of the workstation should display in the **IP Configuration** panel.
• AIX TCP/IP	From the AIX command prompt enter the command **host <hostname>**. The numeric internet address of the workstation identified by <hostname> should appear in the reply.
• HP-UX TCP/IP	From the HP-UX command prompt enter the command **ping <hostname>**. The numeric internet address should appear in the output of the command. Use <Ctrl-c> to end the command.
• Solaris TCP/IP	From the command prompt enter the command **ping - s <hostname>**. The numeric internet address of the workstation identified by <hostname> should appear in the reply.

UNIX users might need to specify a unique port on the workstation to be used for communication, in addition to the internet address. This is necessary because multiple ISPF WSA components can run on any one UNIX workstation at the same time. The port is specified by appending **:<port#>** to the end of the internet address. By default, all of the TCP/IP protocols are defined in the **/etc/services** file. This file can be viewed so you can choose a port that is not currently reserved by another application. Ports 0 through 1024 are reserved for system use. The port number must also be specified in the ISPF WSA graphical user interface. Do this through the **Options** pulldown menu, **Set TCP/IP Port** choice. The port number specified on the host must be the same as the port entered in the ISPF WSA graphical user interface.

The ISPF C/S component on MVS no longer uses the TCP/IP native *IUCV* interface directly for communications. Instead, the C-socket interface is used. C-sockets are enabled by the SAS/C runtime support code distributed with ISPF. The SAS/C runtime supports both the OS/390 Communications Service IP (OE) and common (non-OE) socket implementations enabled by TCP/IP for MVS.

Determining the IBM TCP/IP sockets started task : procedure name

With the enablement of SAS/C socket support the TCP/IP started task procedure name for TCP/IP sockets is determined by reading the TCP/IP configuration data set. The default name for this data set is **TCPIP.TCPIP.DATA**.

The SAS/C runtime code locates the configuration data set by testing for data set access as determined by the elements of the following search order:

1. =TCPIP_DATA=style:any.dataset.name.or.ddname

   =TCPIP_DATA is a SAS/C environment variable. The new SAS/C_TCPIP_DATA_VALUE parameter in the ISPF Configuration table can be used to set the value of this variable. The value **DEFAULT** indicates that no value is set by ISPF.

   Options for setting the value are:

   - DDN:ddname - indicates that SAS/C will search for the data set identified by ″ddname″. If the SAS/C style prefix is omitted the style prefix value is assumed to be ″DDN:″. For example,

     ```
     SAS/C_TCPIP_DATA_VALUE=DDN:MYTCPDD
     SAS/C_TCPIP_DATA_VALUE=MYTCPDD
     (if the TSO logon procedure contains the JCL
     statement //MYTCPDD DD DSN=MYTCPIP.CONFIG ...,
     SAS/C will search for the data set MYTCPIP.CONFIG)
     ```

   - TSO:dataset - indicates that SAS/C will search for the data set ″userid.dataset″ where userid is the TSO userid. For example,

     ```
     SAS/C_TCPIP_DATA_VALUE=TSO:MYTCPIP.CONFIG
     (if the TSO userid is SMITHJ, SAS/C will search
     for the data set SMITHJ.MYTCPIP.CONFIG)
     ```

   - DSN:dataset - indicates that SAS/C will search for the data set ″dataset″.

     ```
     SAS/C_TCPIP_DATA_VALUE=DSN:MYTCPIP.CONFIG
     (SAS/C will search for the data set MYTCPIP.CONFIG)
     ```

2. //SYSTCPD DD or TSO ALLOC F(SYSTCPD)

   //SYSTCPD DD or **TSO ALLOC FI(SYSTCPD)** indicates that SAS/C will search for the data set associated with the ddname ″//SYSTCPD″. For example,

   ```
   //SYSTCPD DD DSN=TCPIP.TCPIP.DATA ...
   (if the TSO logon procedure contains the //SYSTCPD
   DD JCL statement, then SAS/C will search for the
   data set TCPIP.TCPIP.DATA)
   ```

3. userid.TCPIP.DATA

   userid.TCPIP.DATA indicates that SAS/C will search for the data set identified by concatenating the TSO userid to the character string ″.TCPIP.DATA″. If the prefix value set in the user's TSO profile differs from the TSO userid the prefix value will be used instead of the TSO userid. Some examples follow:

   ```
   SMITHJ.TCPIP.DATA
   (if the TSO userid is "SMITHJ" SAS/C will search
   for the dataset SMITHJ.TCPIP.DATA)
   ```

   ```
   JONESB.TCPIP.DATA
   (if the TSO userid is "SMITHJ" and the TSO prefix
   is set to "JONESB" SAS/C will search for the
   dataset JONESB.TCPIP.DATA)
   ```

4. SYS1.TCPPARMS(TCPDATA)

   SYS1.TCPPARMS(TCPDATA) indicates SAS/C will search for the data set SYS1.TCPPARMS and the member TCPDATA.

5. =TCPIP_PREFIX=prefix

=TCPIP_PREFIX is a SAS/C environment variable. The SAS/C_TCPIP_PREFIX_VALUE in the ISPF Configuration table can be used to set the value of the variable. The value ″DEFAULT″ indicates that no value will be set by ISPF. If a string other than ″DEFAULT″ is specified for SAS/C_TCPIP_PREFIX_VALUE, SAS/C will concatenate the string to the characters **.TCPIP.DATA**.

```
SAS/C_TCPIP_PREFIX_VALUE=TCPIP
(SAS/C will search for the data set TCPIP.TCPIP.DATA)
```

6. zap_prefix.TCPIP.DATA

   zap_prefix.TCPIP.DATA - indicates that SAS/C will search for the data set identified by concatenating the SAS/C runtime internal prefix string to the string ″.TCPIP.DATA″. The default value for the prefix string is ″TCPIP″ which implies the data set name TCPIP.TCPIP.DATA. Information on applying a zap to change the prefix to a string other than ″TCPIP″ is available with SAS/C compiler documentation if the SAS/C compiler is installed on your MVS system. If you simply use the SAS/C runtime support supplied with ISPF you should control the search for the TCPIP.DATA set by using one of the other search options.

The elements in the search order list are examined one by one. The search ends when an element in the list enables a data set to be located. That data set is considered to be the TCP/IP configuration data set. The content of the data set is not validated. If the selected data set does not contain valid configuration data for the installed TCP/IP subsystem, TCP/IP socket calls made from the ISPF Client/Server component to the SAS/C runtime might fail and an ISPF Client/Server connection to a workstation might not be possible for TCP/IP.

Search order specification for the TCPIP.DATA data set is currently not required for the ISPF interface to OS/390 Communications Service IP sockets.

Selecting the OS/390 Communications Service IP socket implementation

If the OS/390 Communications Service IP (formerly known as MVS Open Edition) socket implementation is to be used instead of the common TCP/IP socket implementation the ISPF Configuration table keyword USE_MVS_OPEN_EDITION_SOCKETS must be set to YES. The default value NO indicates that the common (non-OE) socket implementation is desired.

Changing ISPF defaults

The ISPF Configuration table must be modified if the value of either of the ISPF defaults keywords is to be changed to a value other than the default value shipped by ISPF. Information regarding customization of the ISPF Configuration table can be found in *ISPF Planning and Customizing*.

TCP/IP additional tips

The ISPF Workstation Agent is not linked with the application programming interface (API) modules provided by any specific communications software vendor. The default behavior for accessing a TCP/IP subsystem in IBM OS/2 environments is for the ISPF WSA to try to locate **tcpipdll.dll** which contains the IBM TCP/IP API modules. If that dynamic link library can be located, then no attempt is made to locate any other TCP/IP dynamic link library.

If the IBM TCP/IP library cannot be located then the workstation agent will try to locate **rcb43.dll** which contains the Novell TCP/IP API modules. If neither

dynamic link library can be located or if the TCP/IP API modules cannot be loaded successfully from a selected library then TCP/IP communication will be inoperative.

The default behavior for accessing a TCP/IP subsystem in Microsoft Windows environments is for the ISPF WSA to try to locate **WINSOCK.DLL**. In the Microsoft Windows environment, many different vendors supply a *winsock.dll* so it is critical that the first *winsock.dll* located by the ISPF WSA contains the TCP/IP API modules actually used by the active TCP/IP subsystem on the workstation. The difficulty of managing multiple *winsock.dll* files in a given workstation environment is compounded by the fact that the search order used by Windows to locate a dynamic link library is not constrained by a *CONFIG.SYS LIBPATH* statement as it is in OS/2 environments. The Windows search order is as follows:

1. current directory
2. *WINDOWS* directory
3. *WINDOWS\SYSTEM* directory
4. directory containing the executable file for the current task
5. directories listed in the *PATH* environment variable
6. list of directories mapped in a network

If the dynamic link library cannot be located or if the TCP/IP API modules cannot be loaded successfully from the selected library then TCP/IP communication will be inoperative.

You have the option to specify an explicit path to the socket DLL used by the active TCP/IP subsystem in a Microsoft Windows workstation environment. The specification of an explicit path overrides the default DLL search order for Windows. The directory defined by the explicit path is searched for WINSOCK.DLL. The explicit path to the TCP/IP socket DLL is specified by using the **Set WINSOCK Path** function available from the **Options** pull-down found on the Client/Server Agent Window (see**The Workstation Agent Window** section of Option 3.7 in the *ISPF User's Guide Volume II*). This function is useful in environments such as those using LAN operating systems, in which the directories containing software from several TCP/IP vendors can be accessed by a workstation.

The Client/Server feature of ISPF (ISPF WSA) takes advantage of the TCP/IP *keepalive* socket option to enable ISPF on the host to detect an abnormal end to a session with the ISPF Workstation Agent on the workstation. Abnormal endings include such events as powering off or rebooting the workstation before closing the session with the ISPF WSA agent. The behavior of the keepalive facility differs for each workstation platform and TCP/IP product supported by ISPF. Some of the differences follow:

Workstation Platform	**Keepalive behavior**
OS/2	Reboot (CTRL-ALT-DELETE), power off, and Workplace Shell shutdown are detected.
Windows	Reboot (CTRL-ALT-DELETE), power off, and Program Manager close are detected.
Windows for Workgroups	Reboot (CTRL-ALT-DELETE), power off, and Program Manager close are detected.
AIX	Reboot, power off, and shutdown of an AIX host are detected. Reboot (CTRL-ALT-BACKSPACE) of

	an X-station client is detected, but power off of an X-station client is detected only when the X-station is powered on again.
HP-UX	Reboot, power off, and shutdown of an HP-UX host are detected.
Solaris	Reboot, power off, and shutdown of a Solaris host are detected.

If no transmissions have been received over a socket connection to the workstation during the specified timer interval, TCP/IP on MVS sends a keepalive packet to the workstation. If there is no response on the socket connection or if the socket connection has been reset, an error is returned and the ISPF C/S session ends on MVS.

The keepalive timer value for IBM TCP/IP for MVS is controlled by the KEEPALIVEOPTIONS statement in the TCP/IP configuration data set. The INTERVAL parameter specifies the number of minutes that TCP/IP waits after the last transmission from the workstation before sending a keepalive packet. The SENDGARBAGE parameter specifies whether the packet contains any data. ISPF C/S was tested with the INTERVAL value set to ***1*** and the SENDGARBAGE value set to *TRUE*.

Enabling the OS/390 Communications Service IP socket implementation

The OS/390 Communications Service IP (formerly known as MVS Open Edition, and called simply OE in this section) started task must be active in order to use ISPF Client/Server with the OE socket implementation. Failure to start OE will result in an SAS/C runtime message such as the following when an ISPF Client/Server session is started from MVS:

```
LSCX902 **** WARNING **** ERRNO = EMVSNOTUP
  Generated in   SOCKET called from line   9512 of @@671470(NF6),
  offset 00020A
  C++ name: DtTCPCLIENT::BeginOperation
  socket failed because the OpenEdition kernel is not running,
  reason code 0000004E.
```

The TSO/ISPF userid must be authorized to use OE services if the OE socket implementation is selected. Failure to obtain proper authorization will result in an MVS abend code ″EC6″ when an ISPF Client/Server connection is attempted. The ″EC6″ abend code indicates a failure of the OE interface. Authorization status for basic OE services can be determined by entering the **omvs** command from the TSO READY prompt. The following example indicates that the TSO user is not authorized to use OE services:

```
 READY
omvs
 FSUM2057I No session was started.  This TSO/E user ID does not have
access to OpenMVS.+
 READY
```

It is not necessary to run the OE shell in order to use the OE socket implmentation with ISPF.

Understanding SAS/C socket TCP/IP error codes

ISPF continues to return a TCP/IP error code in the ″network data″ portion of ISPF messages indicating a TCP/IP communication failure. The TCP/IP error codes returned for failing socket calls are not defined in the IBM MVS TCP/IP documentation. Instead, the error codes are those used by the SAS/C socket

implementation. The codes are found in the SAS/C file **errno.h**. Refer to the *ISPF Messages and Codes* manual for more information.

Configuring APPC connections

The following table provides a fastpath for configuring APPC communications to support ISPF C/S.

APPC Requirements Fast Path

Workstation	MVS host
A fully qualified network name must be defined for a SNA independent LU type 6.2 on the workstation (e.g. USIBMNR.NRI41G00).	The fully qualified network name defined for a SNA independent LU type 6.2 on the workstation must be included in a MVS VTAM definition for the workstation node and must be specified on the *ISPSTART* command or from the GUI settings panel to invoke ISPF WSA.
No workstation requirement matching this host requirement →	Either a *NOSCHED* LU defined with the *BASE* option or a LU defined with the *BASE* and *SCHED(ASCH)* options in the MVS parmlib member *APPCPMxx* is required to support outbound conversation requests from ISPF to the workstation.
No workstation requirement matching this host requirement →	A VTAM application LU must be defined such that the label of the APPL definition statement and its ACBNAME parameter value match the *ACBNAME* of the APPC LU defined in MVS parmlib member *APPCPMxx*.
The mode #*INTER* must be defined on the workstation.	The mode #*INTER* must be defined as an entry in a MVS VTAM mode table associated with the VTAM definition of the workstation node.
No workstation requirement matching this host requirement →	The *SYS1.CSSLIB* data set containing the APPC callable system services modules must be accessible by ISPF.

APPC Requirements Detail

A fully qualified SNA network name is an identifier up to 17 bytes in length assigned to a SNA logical unit (LU) and represented as two symbols, each up to 8 bytes in length, separated by a period. For example:

```
USIBMNR.NRI41G00
```

The symbol preceding the period specifies the name of a SNA network and the symbol following the period specifies the name of an LU within that network. For the purpose of ISPF C/S communications you need to know the fully qualified network name of an independent SNA LU type 6.2 associated with the workstation on which the ISPF WSA graphical user interface is to be displayed. The LU type 6.2 is required by the SNA communications software to support APPC connections to the workstation. For information on installation and customization of the SNA APPC software used on your workstation consult the documentation supplied by your software vendor. If you have already installed SNA APPC communications software on your workstation and you do not know the fully qualified network

name of an independent SNA LU type 6.2 you can identify a candidate name as follows:

If you have this SNA APPC workstation software	do this on the workstation
• OS/2 Access Feature of IBM Communications Server for OS/2 Warp, Version 4	Select the *Command Prompts* folder after selecting the *OS/2 System* icon. Select *OS/2 Window* from the folder and at the displayed OS/2 command prompt enter the command *pmdsplay*. Select *Display* from the *Display Active Configuration* menu followed by the *General SNA* option on the pulldown menu. From the displayed listbox select *Logical unit 6.2* and a list of independent SNA type 6.2 logical units will be displayed each with its associated fully qualified network name.
• IBM APPC Networking Services for Windows 1.00.02 (base product with CSD #2).	From the ″IBM APPC Networking Services″ program group select the icon for the **Configure** program item. Select the **Step 1** pushbutton from the Networking Services Configuration window. The fully qualified local LU name for APPC communications will appear in the General Configuration window.

The fully qualified name of the independent LU selected for the ISPF WSA connection on the workstation must also be identified to VTAM on the MVS system on which the ISPF C/S host agent will be invoked. The workstation independent LU can be identified to VTAM by one of the following methods:

- by dynamic definition by VTAM during session establishment
- by CDRSC definition statements
- by standard LU definition statements with *LOCADDR=0* coded

The *VTAM Network Implementation Guide* and the *VTAM Resource Definition Reference* provide details on each method.

An LU definition must also be associated with the ISPF C/S agent on the MVS host to enable the connection to an ISPF WSA agent on a workstation. This LU must be made available to ISPF C/S by an APPC/MVS definition of a *base* LU to handle outbound connection requests from MVS to the workstation. The LU can be defined in one of two ways in the *APPCPMxx* member of the MVS parmlib data set by using a *LUADD* statement that includes one of the following combinations of parameters:

- the *NOSCHED* and *BASE* parameters
- the *SCHED(ASCH)* and *BASE* parameters

The option involving the *NOSCHED* parameter is not available in MVS releases prior to MVS/ESA SP Version 4.3. If neither LU option is defined outbound session requests from the ISPF C/S agent on the MVS host will be rejected.

A VTAM application LU must be defined in VTAMLST to match the definition in *APPCPMxx*. Both the label of the *APPL* statement and the *ACBNAME*= parameter value must be identical to the value specified for the *ACBNAME* parameter on the *LUADD* statement for the APPC/MVS LU defined in MVS parmlib member *APPCPMxx*.

A logical connection called a *session* must exist between the LU associated with the ISPF C/S agent on the MVS host and the LU associated with the ISPF WSA agent on the workstation to support APPC communication. To establish a session the LU on the MVS host must send a set of session control parameters called a *bind* to the LU on the workstation. The parameters are selected based on a *mode* definition. The ISPF C/S agent on MVS uses the *#INTER* definition which is located in the IBM-supplied default VTAM mode table *ISTINCLM* The *#INTER* mode definition is also included among the default definitions supplied by the supported APPC communications software for workstations.

Beginning with MVS/ESA SP Version 4.3 the *SYS1.CSSLIB* data set is automatically concatenated to *SYS1.LINKLIB* when the MVS system *LNKLST* is created. For prior MVS releases *SYS1.CSSLIB* should be added to the *LNKLST* concatenation defined in the *LNKLSTxx* member in the MVS parmlib. *SYS1.CSSLIB* contains modules that support the APPC/MVS callable services used by the ISPF C/S agent on the MVS system.

The VTAM node definition for the independent LU used by APPC Networking services for Windows cannot be the same VTAM node definition used for your native or emulated 3270 terminal support. A single node definition cannot be shared by APPC Networking Services for Windows and your 3270 terminal support. Likewise, if you are using a 3270 emulator on the same Windows system on which you intend to use APPC Networking Services for Windows you must ensure that different local SNA service access point (SAP) addresses are used by the 3270 emulator and by APPC Networking Services for Windows. To change the local SAP address for APPC Networking Services for Windows select the icon for the **Configure** program item from the IBM APPC Networking Services program group. Select the **Step 2** pushbutton from the Networking Services Configuration window and then select **Advanced** from the LAN Configuration window. The local SAP can be changed on the Advanced LAN Configuration window.

APPC additional tips

To activate APPC Networking Services for Windows and Personal Communications/3270 in any order in that environment it was necessary to increase the values of the MINSAP and MINLINK parameters for the IEEE 802.2 protocol driver for NDIS in the CONFIG.SYS file as follows:

```
DEVICEHIGH = C:\LSP\DXME0MOD.SYS N ,,,2,2
```

where C:\LSP specifies the LAN Support Program directory in which the IEEE 802.2 protocol driver (DXME0MOD.SYS) was installed.

The final two positional parameters specify the minimum service access point and minimum link station values required to support activation of an independent LU 6.2 for APPC Networking Services for Windows and activation of a dependent LU 2 for Personal Communications/3270. If additional LUs are required, for example to support additional dependent LU 2 connections, the minimum service access point and minimum link station values might need to be increased.

Configuration of network support for SNA APPC communications is somewhat more complex than configuration for TCP/IP owing largely to the transition that SNA has been making in recent years from an essentially static hierarchical networking model to a dynamic peer-to-peer networking model. Recognizing the difficulty that this transition presents to customers trying to configure APPC application support and APPN network support, networking specialists at IBM have produced the *Multi-platform APPC Configuration Guide* (GG24-4485). This

guide is a valuable aid for anyone who wants to understand how to configure APPC and APPN communications in any of the environments in which the ISPF C/S capability can be enabled. The guide may be available from one of the following sources:

- Outside IBM:

  CompuServe
  From the APPC Info Exchange (GO APPC)

  MPCONP.ZIP
  The Postscript version of the document

  MPCONT.ZIP
  The ASCII text version of the document

  MPCONB.ZIP
  The BookManager version of the document

  MPCONF.ZIP
  The sample files to be used with the document

  OS2BBS
  From the IBM bulletin board accessible through IIN

  MPCONFIG ZIPBIN
  The Postscript version, the ASCII text version, and the sample files

  MPCONFIG LIST3820
  The MPCONFIG LIST3820 file

  FTP site on Internet
  From anonymous ftp to networking.raleigh.ibm.com (192.35.236.4) in directory /pub/appc_appn

- Inside IBM:

  MKTTOOLA
  From the IBM marketing bulletin board with packages available for distribution to customers.

  MPCONFIG PACKAGE
  contains the Postscript, ASCII test, and LIST3820 version of the document as well as sample files.

  OS2TOOLS
  From the bulletin board for OS/2 tools

  MPCONFIG PACKAGE
  contains the Postscript, ASCII test, and LIST3820 version of the document as well as sample files.

One way that you can simplify APPC configuration is to use the fully qualified network name of the SNA *control point* defined for your workstation as the independent SNA LU type 6.2 that will support ISPF C/S connections. The *control point* for your workstation represents various subsystem functions associated with your communications software. The fully qualified network name for the *control point* can be used not only to identify that set of subsystem capabilities but also to support applications such as the ISPF workstation agent that use an independent SNA LU type 6.2. If you do not know, or cannot remember, the fully qualified name associated with your workstation *control point* you can determine the name as follows:

If you have this SNA APPC workstation software	do this on the workstation
• OS/2 Access Feature of IBM Communications Server for OS/2 Warp, Version 4	Select the *Command Prompts* folder after selecting the *OS/2 System* icon. Select *OS/2 Window* from the folder and at the displayed OS/2 command prompt enter the command *pmdsplay* . Select *Display* from the *Display Active Configuration* menu followed by the *General SNA* option on the pulldown menu. From the displayed listbox select *Global SNA* and an information window will be displayed. The *Network name* together with the *Control point (CP) name* constitute the fully qualified control point name associated with your workstation.
• IBM APPC Networking Services for Windows 1.00.02 (base product with CSD #2).	From the ″IBM APPC Networking Services″ program group select the icon for the **Configure** program item. Select the **Step 1** pushbutton from the Networking Services Configuration window. The fully qualified local LU name for APPC communications will appear in the General Configuration window.

Appendix C. Listing Formats

This appendix describes and displays the kinds of listings you can produce using ISPF. The sample listings shown on the following pages are for illustration purposes only. They are not intended to be exact replicas, because printouts of ISPF listings vary according to the kind of printer you are using.

Source and Index Listings

If autolist mode is on, the ISPF editor automatically generates a source listing when you finish editing. You can also get source listings and index listings by using the following utility options:

Library (3.1)
P - Print member

Move/Copy (3.3)

- CP - Copy and print
- MP - Move and print

Data Set List (3.4)

- P - Print data set list
- PV - Print VTOC entries

Hardcopy (3.6)

- PK - Print and keep data set
- PD - Print and delete data set

Outlist (3.8)
P - Print job output.

Source Listings

Figure 52 on page 190 shows a sample source listing.

```
PROJECT:  ISPFDEMO                 MEMBER:   COINS                 DATE:  86/07/30
GROUP:    MYLIB                    LEVEL:   01.04                  TIME:   17:22
TYPE:     PLI                      USERID:   USERID                PAGE   01 OF 01

START                                                                                     MOD
 COL  ------+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8                 FLAGS

 2   COINS                                                                 00010001       *
 4     PROCRDURE OPTIONS  (MAIN):                                          00020000
 6       DECLARE                                                           00030000
 8         COUNT                                                           00040000
 8         HALVES                                                          00050000
 8         QUARTERS                                                        00060003       *
 8         DIMES                                                           00070000
 8         NICKELS                                                         00080004       **
 8         SYSPRINT                                                        00090000
 6       DO HALVES = 100 TO 0 BY -50;                                      00100000
 8         DO QUARTERS = (100 - HALVES) TO 0 BY -25;                       00110000
10           DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;     00120000
12             NICKELS = 100 - HALVES - QUARTERS - DIMES;                  00130000
12             PUT FILE (SYSPRINT) DATA (COUNT, HALVES, QUARTERS, DIMES, NICKELS); 00140000
12             COUNT = COUNT + 1;                                          00150000
10           END;                                                          00160000
 8         END;                                                            00170000
 6       END;                                                              00180000
 4     END COINS;                                                          00190001       *
```

Figure 52. Sample Source Listing

Information at the top of the page includes project, group, type, and member name, current version and modification level, user ID, date and time that the listing was produced, and page number.

A column-positioning line is printed following the heading and preceding the actual data. The *start column* is printed to the left of each line,indicating the position of the first character in each line that is not a blank.

For ISPF library members with statistics, asterisks are either printed or not printed to the right of each line according to the setting of the modification flag, as follows:

- If the modification flag (columns 79-80) in the line is 00, no asterisks are printed.
- If the modification flag is nonzero but differs from the current modification level of the member, a single asterisk (*) is printed.
- If the modification flag is nonzero and has the same value as the current modification level of the member, two asterisks (**) are printed.

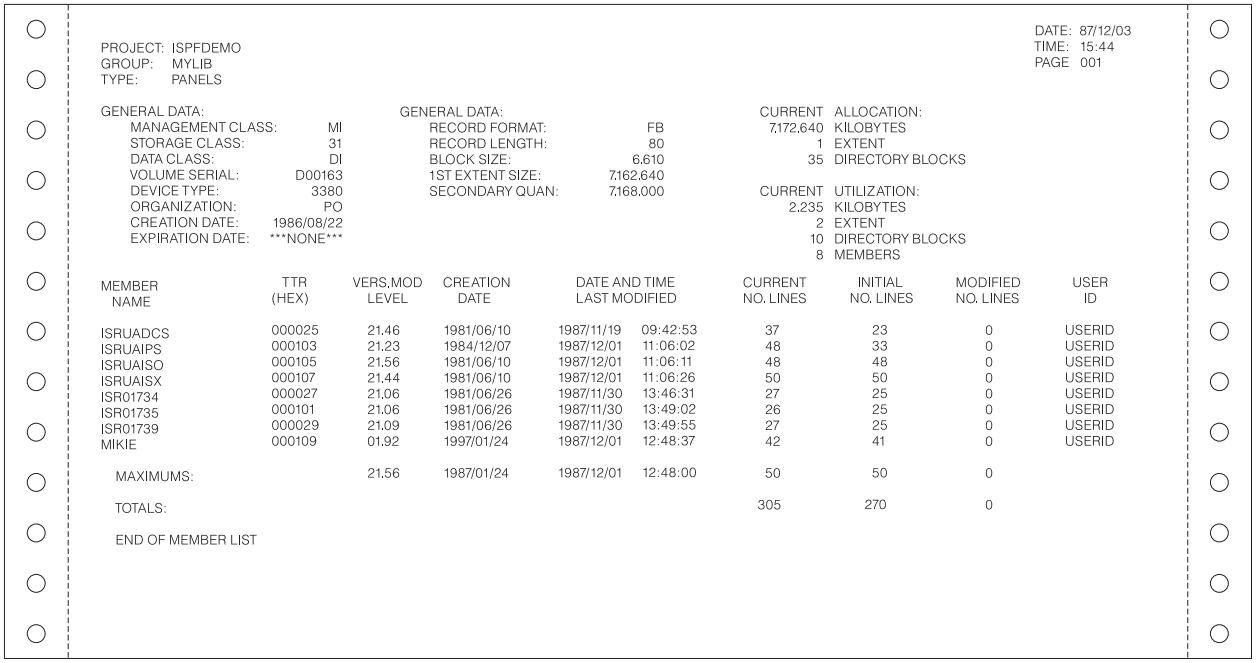
The asterisks allow you to scan the listing quickly for lines that were added or changed since the version was created (*) and for lines that were added or changed during the last update (**).

Index Listings

ISPF provides index listings at your request through the X (Print index listing) option of the Library utility (3.1), or the X or PX options of the Data Set List utility (3.4).

Index Listings for Source Libraries

Figure 53 on page 191 shows a sample index listing for an ISPF library.



DATE: 87/12/03
TIME: 15:44
PAGE 001

PROJECT: ISPFDEMO
GROUP: MYLIB
TYPE: PANELS

GENERAL DATA:
MANAGEMENT CLASS: MI
STORAGE CLASS: 31
DATA CLASS: DI
VOLUME SERIAL: D00163
DEVICE TYPE: 3380
ORGANIZATION: PO
CREATION DATE: 1986/08/22
EXPIRATION DATE: ***NONE***

GENERAL DATA:
RECORD FORMAT: FB
RECORD LENGTH: 80
BLOCK SIZE: 6.610
1ST EXTENT SIZE: 7.162.640
SECONDARY QUAN: 7.168.000

CURRENT ALLOCATION:
7,172,640 KILOBYTES
1 EXTENT
35 DIRECTORY BLOCKS

CURRENT UTILIZATION:
2,235 KILOBYTES
2 EXTENT
10 DIRECTORY BLOCKS
8 MEMBERS

MEMBER NAME	TTR (HEX)	VERS,MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED		CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
ISRUADCS	000025	21.46	1981/06/10	1987/11/19	09:42:53	37	23	0	USERID
ISRUAIPS	000103	21.23	1984/12/07	1987/12/01	11:06:02	48	33	0	USERID
ISRUAISO	000105	21.56	1981/06/10	1987/12/01	11:06:11	48	48	0	USERID
ISRUAISX	000107	21.44	1981/06/10	1987/12/01	11:06:26	50	50	0	USERID
ISR01734	000027	21.06	1981/06/26	1987/11/30	13:46:31	27	25	0	USERID
ISR01735	000101	21.06	1981/06/26	1987/11/30	13:49:02	26	25	0	USERID
ISR01739	000029	21.09	1981/06/26	1987/11/30	13:49:55	27	25	0	USERID
MIKIE	000109	01.92	1997/01/24	1987/12/01	12:48:37	42	41	0	USERID
MAXIMUMS:		21.56	1987/01/24	1987/12/01	12:48:00	50	50	0	
TOTALS:						305	270	0	

END OF MEMBER LIST

Figure 53. Sample Index Listing - Managed Source Library

The sample index listing shown here is for a source library that is managed by the Storage Management Subsystem.

Note: This index listing format is available only when Data Facility Product (DFP) Version 3 Release 1 is installed and available, and when Storage Management Subsystem is active.

The heading information includes:
- Project, group (library), and type
- Date and time the listing was produced
- Page number.

This is followed by general information about the data set, including current space allocation and utilization. The only differences between this index listing and one for a non-managed source library are the following:
- Management, storage, and data classes are shown under the GENERAL DATA heading.
- The 1st extent size, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders.

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a non-managed volume. Following this, the member name and statistics are printed for each member in the data set, arranged in alphabetic order. For sequential data sets, the index listing contains only the general information.

Index Listings for Load Libraries

An index listing for an OBJ library is similar to an index listing for a source library, except that no statistics are maintained. A sample index listing for a LOAD library that is managed by the Storage Management Subsystem is shown in Figure 54.

Note: This index listing format is available only when Data Facility Product (DFP) Version 3 Release 1 is installed and available, and when Storage Management Subsystem is active. See *ISPF User's Guide* for more information.

Here, the module attributes are printed to the right of each member name.

```
                                                                                              DATE:  99/03/15
DATA SET:    ISPF.TEMP.LOAD                                                                   TIME:  10:47
                                                                                              PAGE   001
GENERAL DATA:
     MANAGEMENT CLASS:    STANDARD    GENERAL DATA:                  U          CURRENT  ALLOCATION:
     STORAGE CLASS:           BASE        RECORD FORMAT:             0              608  BLOCKS
     DATA CLASS:          **NONE**        RECORD LENGTH:         6,144                2  EXTENT
     VOLUME SERIAL:         PRM905        BLOCK SIZE:              280               30  DIRECTORY BLOCKS
     DEVICE TYPE:             3390        1ST EXTENT SIZE:         325
     ORGANIZATION:              PO        SECONDARY QUAN:                   CURRENT  UTILIZATION:
     DATA SET NAME TYPE:       PDS                                              500  BLOCKS
     CREATION DATE:     1999/03/15                                                2  EXTENT
     EXPIRATION DATE:   ***NONE***                                                4  DIRECTORY BLOCKS
                                                                                 18  MEMBERS

MEMBER      ALIAS                  SIZE       SIZE      ENTRY     TTR     AUTH                       MODULE
 NAME        OF         SSI        (HEX)      (DEC)     POINT     (HEX)   CODE   AMODE   RMODE   ----ATTRIBUTES----
           ISPSUBS
ISPCIP               D2799050   000D6C00    879,616   0008B420   000020    00      31     ANY      RN  RU
ISPSUBS              D2799050   000D6C00    879,616   00000000   000020    00      31     ANY      RN  RU
ISPVCALL             D5C99036   00009EB0     40,624   00000000   001C0B    00      31      24      RN  RU
ISPVSTG              DF499019   00000C90      3,216   00000000   001D09    00      31      24      RN  RU
ISPVVD               DF499019   00000CB0      3,248   00000000   001D11    00      31      24      RN  RU
ISRLALIN             D5D99036   0000A4D8     42,200   00000000   001D19    00      31     ANY      RN  RU
ISRLCFCL             D2199057   0001B4C8    111,816   00000000   001E13    00      31     ANY      RN  RU
ISRLESS              C4898300   00000F30      3,888   00000000   002112    00      31     ANY      RN  RU
ISRLLDFI             D5D99036   00006F88     28,552   00000000   002119    00      31     ANY      RN  RU
ISRLUACT             D5D99036   00002C00     11,264   00000000   002210    00      31     ANY      RN  RU
ISRPLEX              D5D99036   0000E6C8     59,080   00000000   002303    00      31     ANY      RN  RU
ISRSEPRM             D5D99036   00012B68     76,648   00000000   00240C    00      31     ANY      RN  RU
ISRSFM               D5D99036   000087C0     34,752   00000000   00261B    00      31     ANY      RN  RU
ISRSSM               D5D99036   00009A70     39,536   00000000   002804    00      31     ANY      RN  RU
ISRSUBS              B4299057   000C9148    823,624   00000000   00290C    00      31     ANY      RN  RU
ISRUDA               B4499057   000380D8    229,592   00000000   00160F    00      31     ANY      RN  RU
ISRUDL     ISRUDA    B4499057   000380D8    229,592   0001FAD0   00160F    00      31     ANY      RN  RU
ISRUOLP              D5C99036   00002680      9,856   00000000   003E04    00      31     ANY      RN  RU

   TOTALS:                      00249548  2,397,512

  END OF MEMBER LIST

------

MODULE ATTRIBUTE CODES:
        NX          NOT EXECUTABLE
        OL          DATA OLY, NOT LOADABLE
        OV          IN OVERLAY STRUCTURE
        RF          REFRESHABLE
        RN          REENTERABLE
        RU          REUSABLE
        TS          MODULE CONTAINS TEST INFORMATION
        SC          SCATTER LOAD
```

Figure 54. Sample Index Listing - Managed Load Library

The only differences between this index listing and one for a non-managed LOAD library are the following:

- Management, storage, and data classes are shown under the GENERAL DATA heading.
- The 1st extent size, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders.

As in the source library index listing, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a non-managed volume.

ISPF Log Listings

Figure 55 shows a sample ISPF log listing. The log contains a message for each significant user action, such as saving edited data, moving members from one data set to another, or submitting a batch job.

```
TIME              * * * ISPF TRANSACTION LOG * *                    USERID: USERID   DATE: 86/06/29  PAGE: 1

17:11        START OF ISPF SESSION LOG #64 ----------------------------------------------------------------------
17:12             UTILITY  - ALLOCATE       - ISPFDEMO.NEWLIB.PLI ALLOCATED ON VOLUME WORK 97
17:13             UTILITY  - MOVE/COPY      - ISPFDEMO.MYLIB.PLI (ACCOUNT) COPIED
17:13                                           TO ISPFDEMO.NEWLIB.PLI (ACCTXX)
17:13             UTILITY  - MOVE/COPY      - ISPFDEMO.MYLIB.PLI (ACCT1) COPIED
17:13                                           TO ISPFDEMO.NEWLIB.PLI (ACCTYY)
17:13             UTILITY  - MOVE COPY      - ISPFDEMO. MYLIB.PLI (ACCT2) COPIED
17:13                                           TO ISPFDEMO,NEWLIB.PLI (ACCTZZ)
17:13             UTILITY  - MOVE COPY      - ISPFDEMO.MYLIB.PLI (COINS) COPIED
17:13                                           TO ISPFDEMO.NEWLIB.PLI (COINS)
17:14             EDIT     - SAVE           - ISPFDEMO.NEWLIB.PLI (COINS) - MEMBER SAVED
17:15             EDIT     - CREATE         - ISPFDEMO.NEWLIB.PLI (ACCTNEW) - MEMBER CREATED
17:15             EDIT     - SAVE           - ISPFDEMO.NEWLIB.PLI (ACCTXX) - MEMBER SAVED
17:15             EDIT     - SAVE           - ISPFDEMO.NEWLIB.PLI (ACCTYY) - MEMBER SAVED
17:16             EDIT     - SAVE           - ISPFDEMO.NEWLIB.PLI (ACCTZZ) - MEMBER SAVED
17:17             UTILITY  - DELETE         - ISPFDEMO.NEWLIB.PLI (ACCTZZ)  DELETED
17:17             TSO      - COMMAND        - LISTC
17:17             UTILITY  - ALLOCATE       - ISPFDEMO.NEWLIB.OBJ ALLOCATED ON VOLUME WORK98
17:19             JOB      - USER8          - PLI OPT COMPILE - - TO ISPFDEMO.NEWLIB.OBJ (COINS)
17:19             JOB      - USER8          - SUBMITTED
17:20        END OF ISPF SESSION LOG #64----------------------------------------------------------------------
```

Figure 55. Sample ISPF Log Listing

Member List Listings

This section shows samples of member list listings created by the SAVE command. With this command, you can create listings for both source and load libraries.

The following sample listings show the format used when you do not specify a list ID.

Member List Listings for Source Libraries

Figure 56 on page 194 shows a sample member list listing for a source library. These listings contain the relative block address of each member, shown in hexadecimal format, and other characteristics when available.

DATA SET: ISPF.PRIVATE.PLS

DATE: 99/03/15
TIME: 10:53
PAGE 001

MEMBER NAME	TTR (HEX)	VERS.MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED	CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
$FORMAT	006A01	01.00	1997/06/03	1997/06/03 10:36:39	3376	3376	0	PDFTOOL
ACCTEX2	005E05	01.00	1997/10/09	1997/10/09 13:06:00	46	43	3	P020136
ALLMEMS	002B03	01.00	1997/10/09	1997/10/09 13:06:00	117	112	20	P020136
ALLMEM2	001C06	01.00	1997/10/09	1997/10/09 13:06:00	135	135	0	P020136
APAR3	005501	01.00	1997/10/09	1997/10/09 13:06:00	250	251	0	P020136
BINDER	003501	01.06	1994/02/17	1994/02/17 15:20:38	74	46	44	P020136
CALLER	001B01	01.04	1992/01/30	1992/01/30 12:41:18	165	164	7	TSTLMF4
EXPDATE1	013208	01.00	1998/11/10	2005/05/05 12:12:00	1	1	0	PDFTOOL
ISRCNFIG	009F01							
END OF MEMBER LIST								

Figure 56. Sample Member List Listing for a Source Library

Member List Listings for Load Libraries

Figure 57 shows a sample member list listing for a source library. These listings contain the size of each load module, shown in hexadecimal format, and other characteristics when available.

DATA SET: ISPF.TEMP.LOAD

DATE: 99/03/15
TIME: 10:52
PAGE 001

MEMBER NAME	ALIAS OF	SSI	SIZE (HEX)	SIZE (DEC)	ENTRY POINT	TTR (HEX)	AUTH CODE	AMODE	RMODE	MODULE ----ATTRIBUTES----
ISPCIP	ISPSUBS	D2799050	000D6C00	879,616	0008B420	000020	00	31	ANY	RN RU
ISPSUBS		D2799050	000D6C00	879,616	00000000	000020	00	31	ANY	RN RU
ISPVCALL		D5C99036	00009EB0	40,624	00000000	001C0B	00	31	24	RN RU
ISPVSTG		DF499019	00000C90	3,216	00000000	001D09	00	31	24	RN RU
ISPVVD		DF499019	00000CB0	3,248	00000000	001D11	00	31	24	RN RU
ISRLALIN		D5D99036	0000A4D8	42,200	00000000	001D19	00	31	ANY	RN RU
ISRLCFCL		D2199057	0001B4C8	111,816	00000000	001E13	00	31	ANY	RN RU
ISRLESS		C4898300	00000F30	3,888	00000000	002112	00	31	ANY	RN RU
ISRLLDFI		D5D99036	00006F88	28,552	00000000	002119	00	31	ANY	RN RU
ISRLUACT		D5D99036	00002C00	11,264	00000000	002210	00	31	ANY	RN RU
ISRPLEX		D5D99036	0000E6C8	59,080	00000000	002303	00	31	ANY	RN RU
ISRSEPRM		D5D99036	00012B68	76,648	00000000	00240C	00	31	ANY	RN RU
ISRSFM		D5D99036	000087C0	34,752	00000000	00261B	00	31	ANY	RN RU
ISRSUBS		D5D99036	00009A70	39,536	00000000	002804	00	31	ANY	RN RU
ISRUDA		B4299057	000C9148	823,624	00000000	00290C	00	31	ANY	RN RU
ISRUDL	ISUDA	B4499057	000380D8	229,592	00000000	00160F	00	31	ANY	RN RU
ISRUOLP		B4499057	000380D8	229,592	0001FAD0	00160F	00	31	ANY	RN RU
END OF MEMBER LIST		D5C99036	00002680	9,856	00000000	003E04	00	31	ANY	RN RU

Figure 57. Sample Member List Listing for a Load Library

Formats for Member List Listings

The following is the format used by the SAVE command to create a member list listing for a source library. The members of a source library have formatted records (RECFM≠U).

Table 12. Format of Source Library Member List Listing

Starting Column	Length in Characters	Description
4	8	Member name
19	6	Relative block address in hexadecimal format
25	2	Version number
28	2	Modification level
31	8	Creation date
40	8	Date last modified
49	5	Time last modified
55	5	Current number of lines
61	5	Initial number of lines
67	5	Number of modified lines
73	7	User ID

The following is the format used by the SAVE command to create a member list listing for a load library. The members of a load library have unformatted records (RECFM=U).

Table 13. Format of Load Library Member List Listing

Starting Column	Length in Characters	Description
4	8	Member name
24	6	Load module size in hexadecimal format
33	6	Load module relative block address in hexadecimal format
40	8	Alias
49	2	Authorization code
53	3	Addressing mode
56	3	Residency mode
61	18	Load module attributes

Data Set List Listings

The sample listing in Figure 58 on page 196 shows the format used when you do not specify a data set list ID.

Data Set List Listings

LISTING OF DATA SETS

DATE: 87/04/14
TIME: 28:06
PAGE 1

DATA SET NAME	VOLUME	ORG	RECFM	LRECL	BLKSZ	TRKS	$USED	XT	CREATED
USERID.CLIST	HSMP0C	PO-E	VB	255	6160	1	100	1	1985/10/03
USERID.DB2X.DXT210.DVRIMEXE	HSMP0F	PO	FB	80	3120	15	6	1	1987/03/05
USERID.DB2X.DXT210.DVRJEDIE	HSMP0F	PO	FB	30	6160	15	13	1	1987/03/05
USERID.EXAMPLE.DATASETS	HSMP07	PS	FB	132	6072	19	5	1	1987/04/14
USERID. FWB.CLIST	MIGRAT								
USERID.FWB.SCRIPT	HSMP07	PO	VB		3120	1	100	1	1986/03/19
USERID.FWB.TABLES	MIGRAT			30					
USERID.ISPTABL	HSMP0F	PO	FB		3120	1	100	1	1987/02/02
USERID.LOG.MISC	MIGRAT			80					
USERID.MASTER.ISPROF	MIGRAT								
USERID.PRIVATE.CLIST	HSMP0C	PO-E	VB		6160	1	100	1	1986/10/03
USERID.PRIVATE.LOAD	HSMP0C	PO	U		6144	1	100	1	1986/10/03
USERID.PRIVATE.MSGS	HSMP0C	PO	FB	255	6160	1	100	1	1986/10/03
USERID.PRIVATE.PANELS	HSMP0C	PO	FB	0	6160	2	100	1	1986/10/03
USERID.PRIVATE.SKELS	HSMP0C	PO	FB		6160	1	100	1	1986/10/03
USERID.PRIVATE.TABLES	HSMP0C	PO	FB	30	6160	2	100	1	1986/10/03
				80					
USERID.SMALL.FIXED	HSMP0B	PO	FB	30	600	1	100	1	1987/04/09
USERID.TEST.CLIST	MIGRAT			80					
USERID.TEST.MSGS	MIGRAT								
USERID.TEST.SCRIPT	MIGRAT			60					
USERID.TEST1.SCRIPT	MIGRAT								
USERID.TEST2.SCRIPT	MIGRAT								

Figure 58. Sample Data Set List Listing

Format for Data Set List Listings

Table 14 shows the format of the data set list written by the SAVE command when a data set list ID is specified.

Table 14. Format of Data Set List Listing

Starting Column	Length in Characters	Description
1	44	Data set name
46	7	Volume and volume indicator
53	4	Data set organization
58	5	Data set record format
64	5	Data set logical record length
70	5	Data set block size
76	6	Data set size in tracks
83	3	Percentage of used tracks or pages (PDSE)
87	3	Number of extents used
91	8	Device type
100	10	Creation date
111	10	Expiration date
122	10	Last reference date

Appendix D. APL and TEXT Character Conversion

You can use APL keyboards for all models of 3270 terminals, and TEXT keyboards for 3278 and 3279 terminals with ISPF. Figure 59 and Figure 60 on page 198 show that the 2-byte transmission codes for APL and TEXT characters are converted by ISPF into one-byte codes for internal storage.

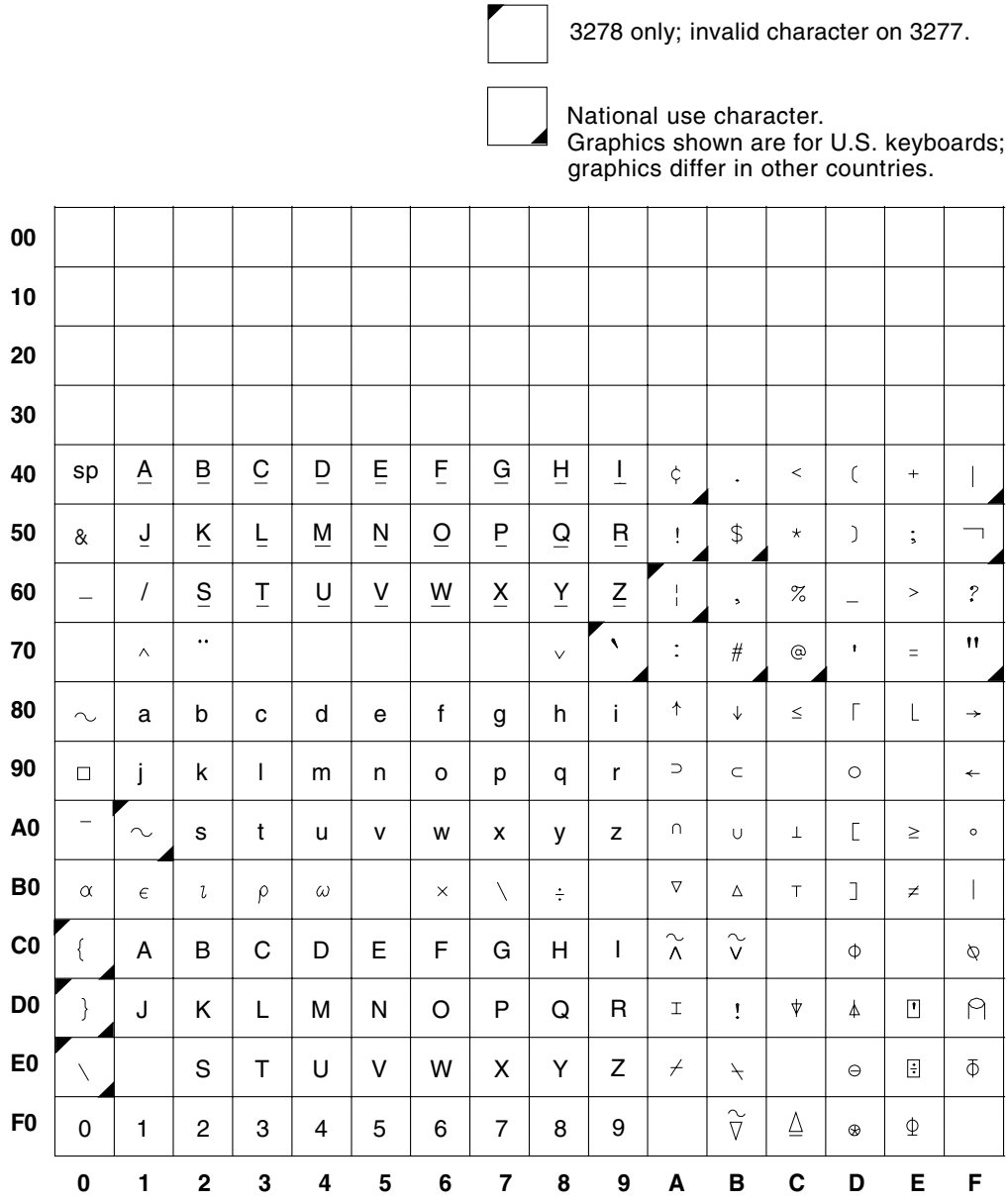


Figure 59. Internal Character Representations for APL Keyboards

National use character.
Graphics shown are for U.S. keyboards;
graphics differ in other countries.

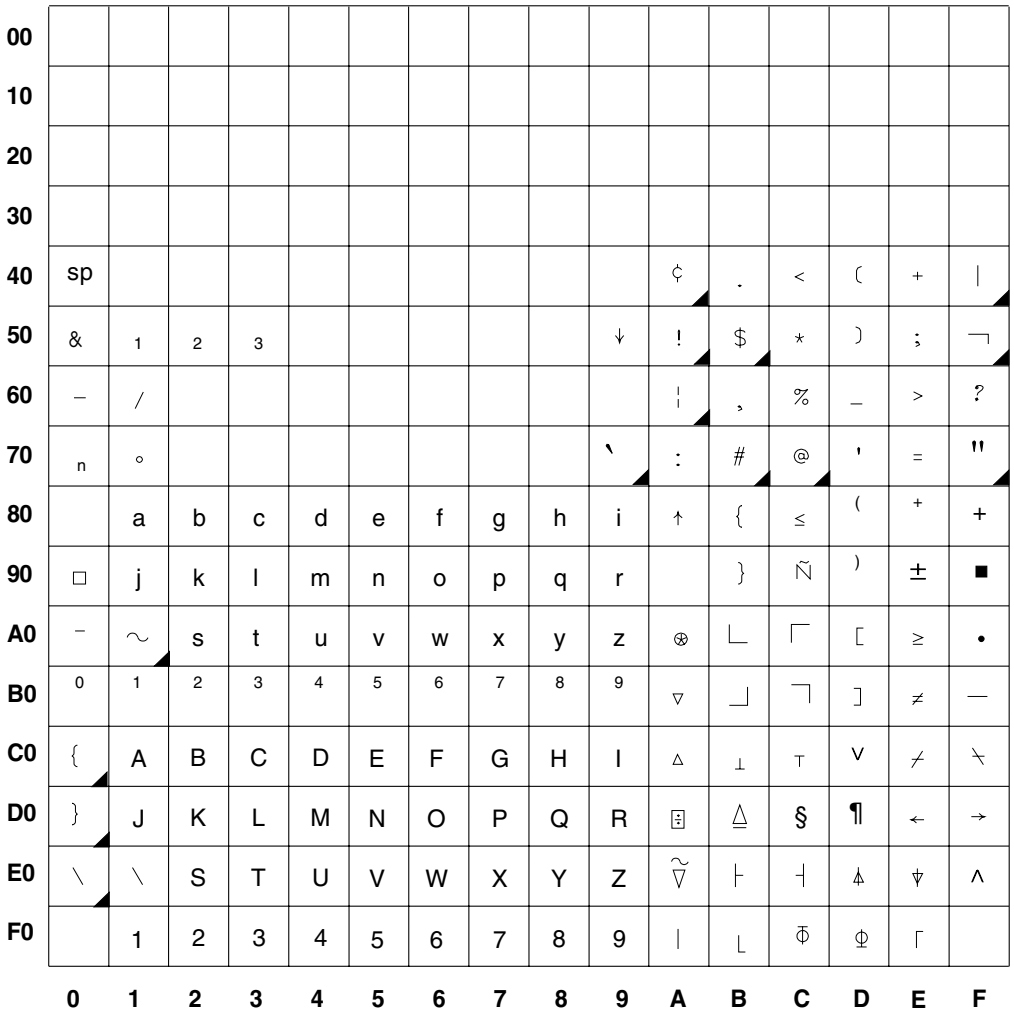


Figure 60. Internal Character Representations for TEXT Keyboards

Appendix E. Abbreviations for Commands and Other Values

The following list includes the commands, field values, keywords/operands, and scroll amounts that can be abbreviated, followed by the allowable abbreviations. To improve readability, you should avoid using abbreviations in edit macros.

Command Abbreviations

BACK	B		
BROWSE	BRO		
CANCEL	CAN		
COLUMNS	COLS	COL	
CONFIRM	CON	C	
DISPLAY	DISPL	DISP	DIS
FIND	F		
INDEX	I		
LOCATE	LOC	L	
RESUME	RES		
SELECT	SEL	S	
SHOWCMD	SHOW		
SKIP	S		
TOP	T		
UP	U		

Field Value Abbreviations

%USED	%		
AFTER	AFT	A	
ALIAS-OF	ALIAS		
ATTRIBUTES	ATTR		
BEFORE	BEF	B	
BLOCK	BLKS		
BOTTOM	BOT	B	
CHANGED	CHA	CHG	C
CREATED	CRE		
CYLINDER	CYLS		
DEVICE	DEV		
EXPIRES	EXP		
GET	G		
MESSAGE	MES		
NO	N		
PUT	P		
REFERRED	REF		
TOP	T	0	
TRACK	TRKS		
VOLUME	VOL		
YES	Y		

Keyword/Operand, Scroll Amount, and Programming Language Abbreviations

CHANGE	CHA	CHG	C
CHARS	CHAR		
COBOL	COB		
DISPLAY	DISPL	DIS	

Keyword/Operand Abbreviations

ERROR	ERR	
LABELS	LABEL	LAB
PREFIX	PRE	
SPECIAL	SPE	
STANDARD	STD	
SUFFIX	SUF	
VERTICAL	VERT	
VOLUME	VOL	

Scroll Amount Abbreviations

CUR	CSR	C
DATA	D	
HALF	H	
MAX	M	
PAGE	P	

Programming Language Abbreviations

ASSEMBLER	ASSEMBLE	ASSEM	ASM
COBOL	COB		
FORTRAN	FORT	FOR	FTN
PASCAL	PAS		
PLI	PLIOPT	PL1	
SCRIPT	SCR	TEXT	

Appendix F. Allocation Data Sets

ISPF issues ALLOC commands based on the ISPF libraries, data set names, list IDs, options, and additional input libraries you enter on the Foreground Assembler H and Foreground VS COBOL II Compile panels. All allocations are done before Assembler H and the VS COBOL II compiler are called.

Because Assembler H and VS COBOL II do not provide a language prompter, ISPF allocates the required data sets for you. The following sections describe the data sets ISPF allocates when you use Assembler H or the VS COBOL II compiler.

SYSIN Data Set

The SYSIN data set is the main input into Assembler H and VS COBOL II. It contains the ISPF libraries or other partitioned or sequential data sets that you enter on the Foreground Assembler H and VS COBOL II Compile panels. This data set is used to find the member that contains the program to be assembled or compiled. For a PDS, the ALLOC command would be:

```
ALLOC FI(SYSIN) DA('proj.lib.type(mem)')
```

For a sequential data set, the ALLOC command would be:

```
ALLOC FI(SYSIN) DA('proj.lib.type')
```

In both commands, `lib` is the library in which the member or data set was found.

SYSLIB Data Set

The SYSLIB data set contains the ISPF library concatenation sequence used to resolve any copy statements specified in your program. It contains the ISPF libraries or other partitioned or sequential data sets and the additional input libraries you specify on the Foreground Assembler H and VS COBOL II Compile panels. For example:

```
ALLOC FI(SYSLIB) DA('SYS1.MACLIB','proj.lib1.type',...,
       'proj.lib4.type','additional lib1','additional lib2')
```

SYSPRINT Data Set

The SYSPRINT data set contains the generated output listing. The entry in the List ID field determines the destination of the output listing. If you enter a name in the List ID field, the output listing is stored in a sequential data set:

```
ALLOC FI(SYSPRINT) DA('prefix.listid.LIST')
```

where `listid` is the name entered in the List ID field. However, if you leave the List ID field blank, ISPF uses the name of the member being assembled or compiled instead of the list ID:

```
ALLOC FI(SYSPRINT) DA('prefix.member.LIST')
```

If you enter an asterisk (*) in the List ID field, ISPF displays the output listing at your terminal, using the following command:

```
ALLOC FI(SYSPRINT) DA(*)
```

SYSPRINT Data Set

See the **List Data Sets** section of the ***Foreground (Option 4)*** chapter of the *ISPF User's Guide Volume II* for more information.

SYSTERM Data Set

The SYSTERM data set contains a summary of the information in the listing data set (SYSPRINT). It is displayed at the terminal if the TERM option is used:

```
ALLOC FI(SYSTERM) DA(*)
```

SYSLIN Data Set

The SYSLIN data set must be preallocated before running Foreground (option 4) or Batch (option 5). The SYSLIN data set contains the object module. This object module will be the input when you link edit. For a PDS, the ALLOC command would be:

```
ALLOC FI(SYSLIN) DA('proj.lib1.OBJ(mem)')
```

For a sequential data set, the ALLOC command would be:

```
ALLOC FI(SYSLIN) DA('proj.lib1.OBJ')
```

SYSPUNCH Data Set

The SYSPUNCH data set is the same as the SYSLIN data set. ISPF does not use this data set. The DUMMY parameter on the ALLOC statement means it should not be used:

```
ALLOC FI(SYSPUNCH) DUMMY
```

SYSUT1 Data Set

The SYSUT1 data set is a temporary utility data set used during processing. It is deleted after it is used.

For Assembler H, the format is:

```
ALLOC FI(SYSUT1) UNIT(SYSDA) NEW DELETE
```

For VS COBOL II, the format is:

```
ALLOC FI(SYSUT1) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
```

SYSUT2—SYSUT7 Data Sets

The SYSUT2, SYSUT3, ..., SYSUT7 data sets are temporary utility data sets used by VS COBOL II only during processing. They are deleted after they are used:

```
ALLOC FI(SYSUT2) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT3) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT4) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT5) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT6) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT7) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
```

Note: SYSUT6 and SYSUT7 are required only if VS COBOL II Version 1, Release 3 is being used.

Appendix G. ISRDDN Diagnostic Utility

ISRDDN is a utility that assists IBM support in evaluating and solving problems. It provides a list of allocated DD names, a list of system ENQs, a list of data sets causing system ENQ contention, and a means of viewing storage within a TSO user's address space. ISRDDN also provides some facilities for gathering information about your system environment.

You can start ISRDDN by issuing the commands TSO ISRDDN or DDLIST from any ISPF command line.

The allocated DD name list shows you all of the DD names allocated to your TSO session. From the list you can perform functions such as Edit or Compress against individual data sets, DD names, or sets of DD names. You can also perform actions against the entire list of displayed DD names.

The ENQ list, available by typing ENQ on the allocation list command line, shows you ENQs on your system. You can limit the size of the list by specifying the QNAME, RNAME, job, user or address space name, and system name.

The ENQ contention list, available by typing CON on the allocation list command line, shows you ENQ contentions on your system for data sets (QNAME SYSDSN).

You can Browse storage using the BROWSE primary command from the allocation list. You can only browse storage which an unauthorized program can see (private and common).

ISRDDN can be used to manipulate the data sets that are allocated, but it also provides the ability to answer questions like:

- Where did a module the user has loaded come from?
- What data sets contain a specific member?
- Are the I/O errors and ABENDs the user is getting due to mixed record format allocations?
- Who is currently allocated to ″SYS1.BRODCAST″?
- What member names or LPA load modules are duplicated in the user's current allocations?
- How many members are in the allocated libraries and which libraries are empty?
- Is the user running ISPF out of LPA or from STEPLIB?

To invoke the ISRDDN program, type **TSO ISRDDN** on any ISPF command line.

The Current Data Set Allocations List

When you start ISRDDN, the ***Current Data Set Allocations*** list displays, as shown in Figure 61 on page 204.

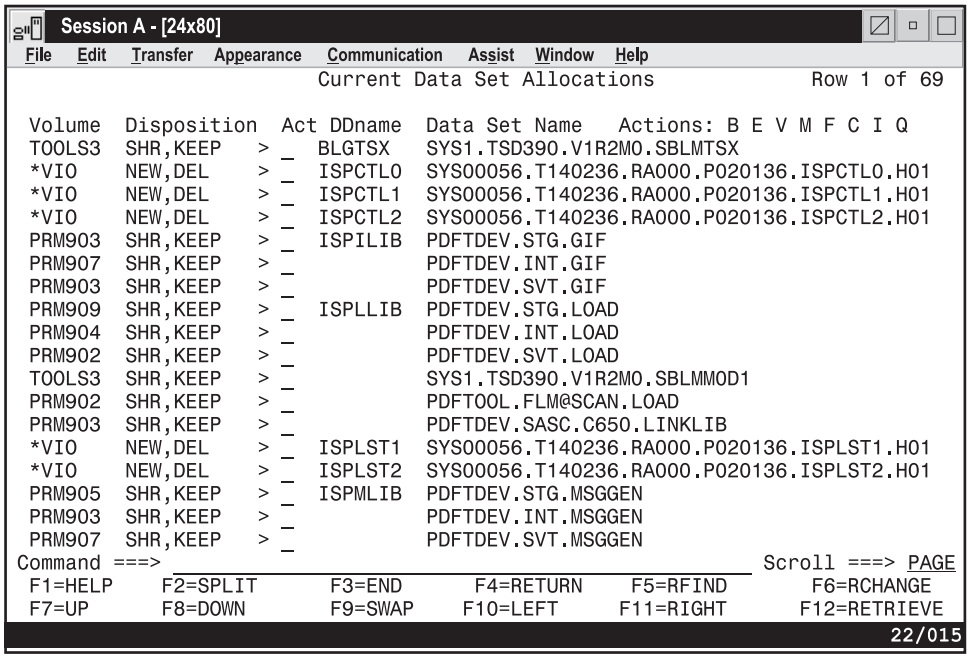


Figure 61. Current Data Set Allocations List panel

On the right side of the display is a list of DD names and their associated data sets. The list of data sets can also contain indicators of DUMMY allocations, subsystem files, or allocations to the terminal. The DD name is shown in white, unless the first data set in the concatenation is scrolled off the top of the screen. If the first data set in a concatenation is not on the screen, the DD name is shown in yellow.

In the center of the display is a column of 1-character input fields, preceded by greater-than signs (>). These input fields are used for line commands such as **E** for Edit or **I** for Information.

The left side of the display contains columns of information about individual data sets. When you scroll right or left, the left side of the screen changes. Initially, the left side of the screen contains the volume name and disposition. If the disposition is red, there are other jobs waiting to use this data set as shown. You can use the **Q** line command to see what jobs are waiting. You can view the VTOC information for a volume by placing the cursor on the volume name and pressing the Enter key.

ISRDDN automatically checks for mixed concatenations when it is started. If you have concatenations of mixed data set types or formats, you are shown a message to that effect when you press the Enter key or scroll the first time. ISRDDN also checks for mixed concatenations when you use the RESET command.

If you scroll right once, you see the attributes of each data set, as shown in Figure 62 on page 205.

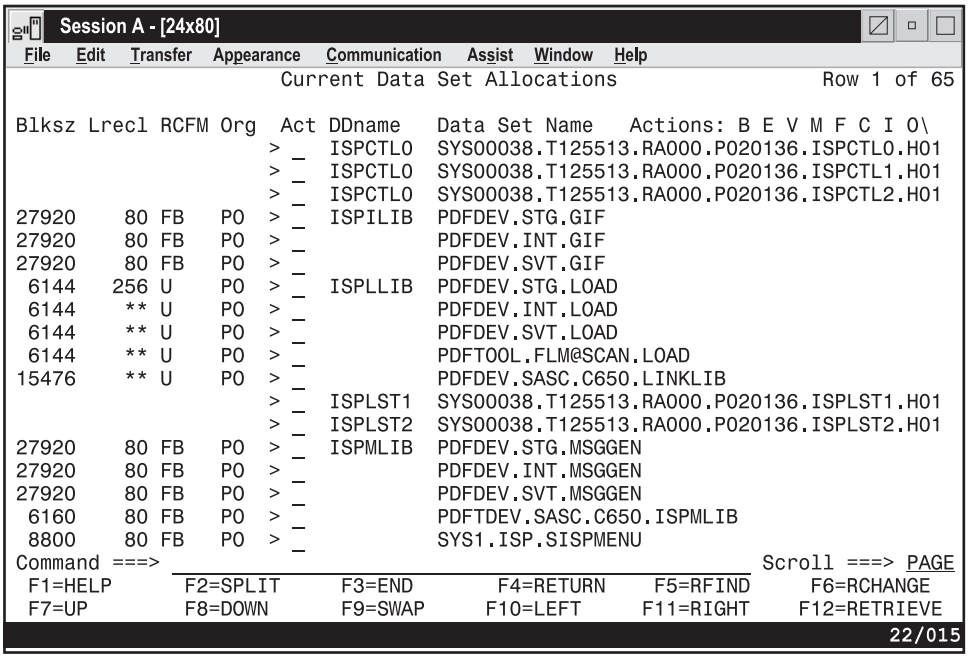


Figure 62. Data Set Attributes in ISRDDN

For some types of allocations, such as subsystem allocations, you might see different information. If you have mixed concatenations, a message with this information appears when you press the Enter key or scroll the first time. You can suppress this message for future innovations of ISRDDN by using the CHECK OFF command

If you scroll right a second time, you see information that includes whether the DD name is open and if so, by how many active DCBs, as shown in Figure 63 on page 206.

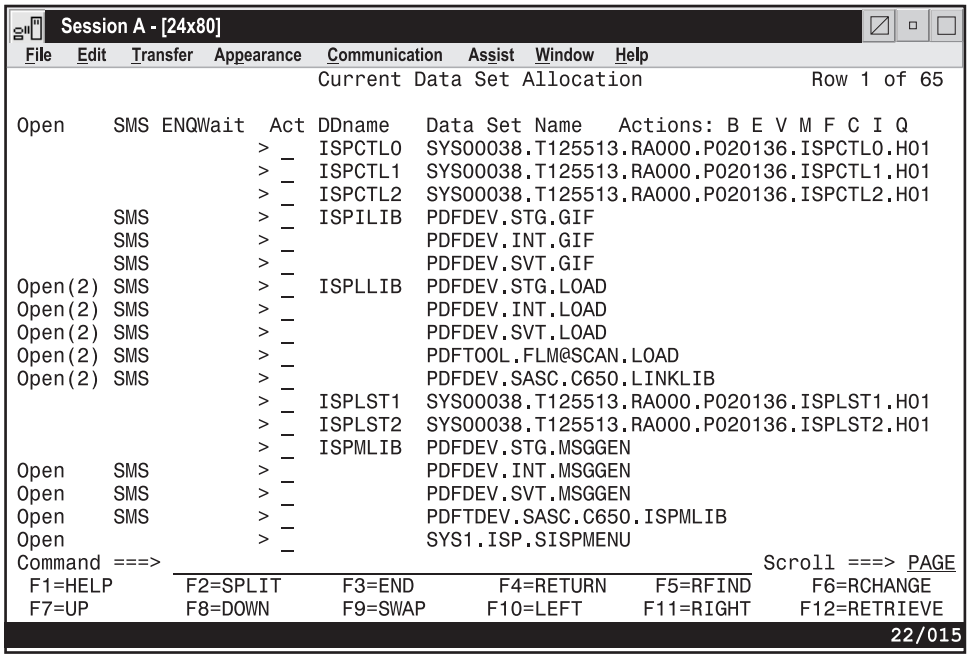


Figure 63. Additional DD Information

You also see the indicator ***SMS*** if the data set is SMS-managed, and information about jobs waiting on the resource. For JES files you might see additional information such as the class and the writer name.

Some primary commands, MEMBER and COUNT, for example, put messages in a fourth status screen (Figure 64.

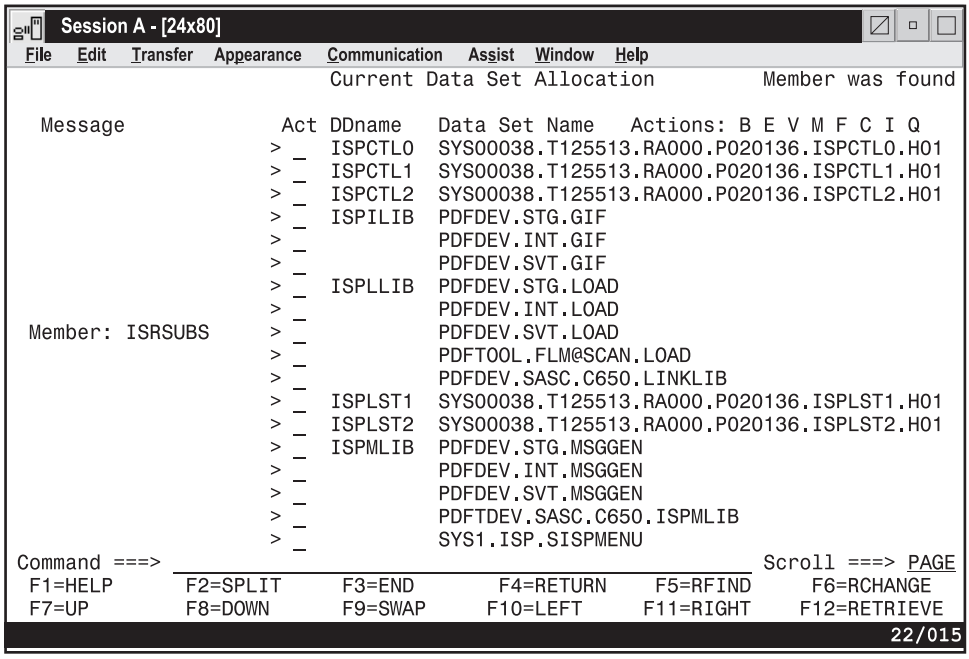


Figure 64. Additional DD Information

If messages exist and you scroll right again, you see the messages. The message screen is only shown if messages exist. If messages do *not* exist, a third scroll to the right returns you to the initial screen.

Using Commands on the Displayed List

The Current Data Set Allocations list supports both primary commands and line commands. The ***displayed list*** is the list of DD names that you can see by scrolling up and down. You can use primary commands to limit what is displayed in the list. Many of the primary commands work only on the contents of the displayed list.

ISRDDN can also create pseudo-DD names that show useful data set names. For example, the **LPA** command adds two pseudo-DD names, LINKLIST and LPALIB, which contain lists of the current link list and LPA libraries.

Allocation List Primary Commands

Primary commands are used to limit the contents of the displayed list, to add pseudo-DD names, to operate on all the contents of the displayed list and to invoke other ISRDDN options.

All primary commands can be invoked with their minimum unique names. For example, MEMBER can be abbreviated as M, while CLIST can be abbreviated as CL. The allocation list primary commands follow.

You can specify an initial primary command when you start ISRDDN. For example, if you enter `DDLISTB 10.???` on an ISPF command line, you will immediately browse the storage containing the TCB control block. When you exit the Browse screen, you are not returned to the DD allocation list. This feature is useful for calling ISRDDN from within a program when, for example, you want to limit the list to specific dd names, view ENQs, save the current allocations, or browse storage.

Only (O) and Exclude (EX, X)

ONLY and EXCLUDE are used to limit the DD names in the displayed list. They take one operand—a whole or partial DD name. For example, the command **O PLI** causes the list to contain only DD names that contain the string ″PLI″, such as STE**PLI**B and ISP**PLI**B.

The ONLY and EXCLUDE commands are useful when you want to limit the DD names or pseudo-DD names that are operated on by commands like MEMBER and DUPLICATES. They are also helpful in reducing the size of the displayed list for easier viewing.

Find (F) and Locate (L)

FIND and LOCATE search the list for a string. LOCATE looks only at DD names and always locates the first matching DD name. FIND looks at everything currently in the displayed list and finds the next occurrance of the string following the current cursor position. You can repeat a FIND opeartion by pressing the RFIND key.

When a string is found by FIND, the string is highlighted and the cursor is placed on the string. When a string is found by LOCATE, the string is highlighted and the cursor is placed in the line command area next to the located DD name.

Reset (R)

The RESET command rebuilds the list. In most screen formats the list is automatically rebuilt when you press Enter. However, if you have used the COUNT command or the MEMBER command and have messages showing in the list, you might need to use the RESET command to refresh the list.

Short (S) and Long (LON)

The SHORT and LONG commands alter the format of the list. The SHORT command places the DD name of a concatenation next to the first data set (as shown in Figure 61 on page 204). The LONG command formats the list with DD names of concatenations placed on a separate line before the data set names (as shown in Figure 65).

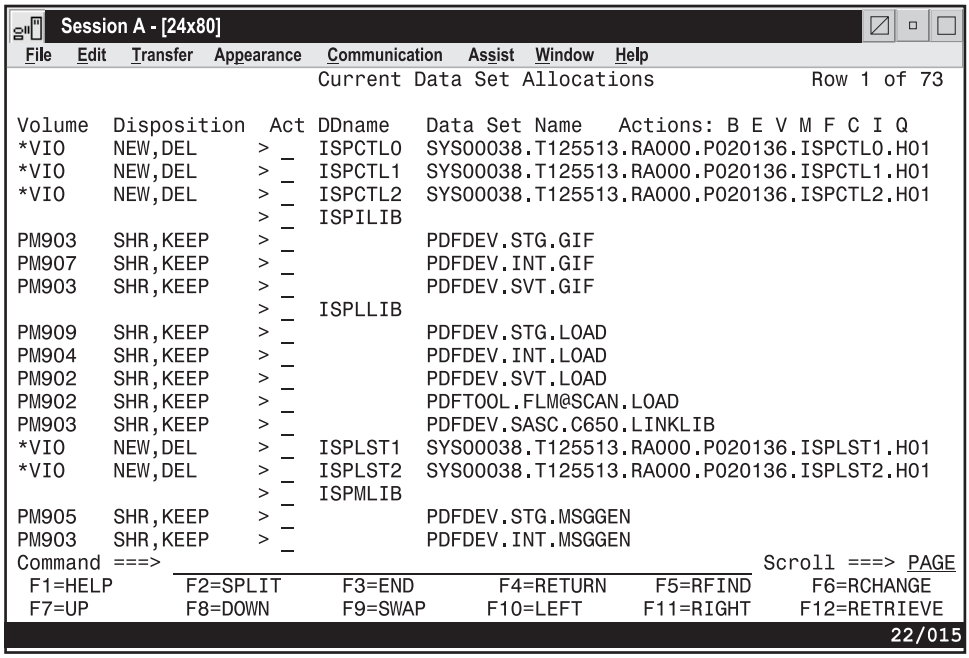


Figure 65. Current Data Set Allocations List in LONG Format

The SHORT format shows more information on one screen. Use the LONG format when you want to use line commands that operate on whole concatenations, such as **E** and **V**, on only the first data set in a concatenation.

Member (M)

The MEMBER command is a very useful command in ISRDDN. MEMBER searches the displayed list (or just DD names containing a given string) for a member whose name matches a pattern. For example, the command **M ISRSUBS** searches the data sets in the displayed list, the job pack area, and the link pack directory for members named ISRSUBS. Data sets that contain the member are flagged with a message on the left side of the list, as shown in Figure 66 on page 209.

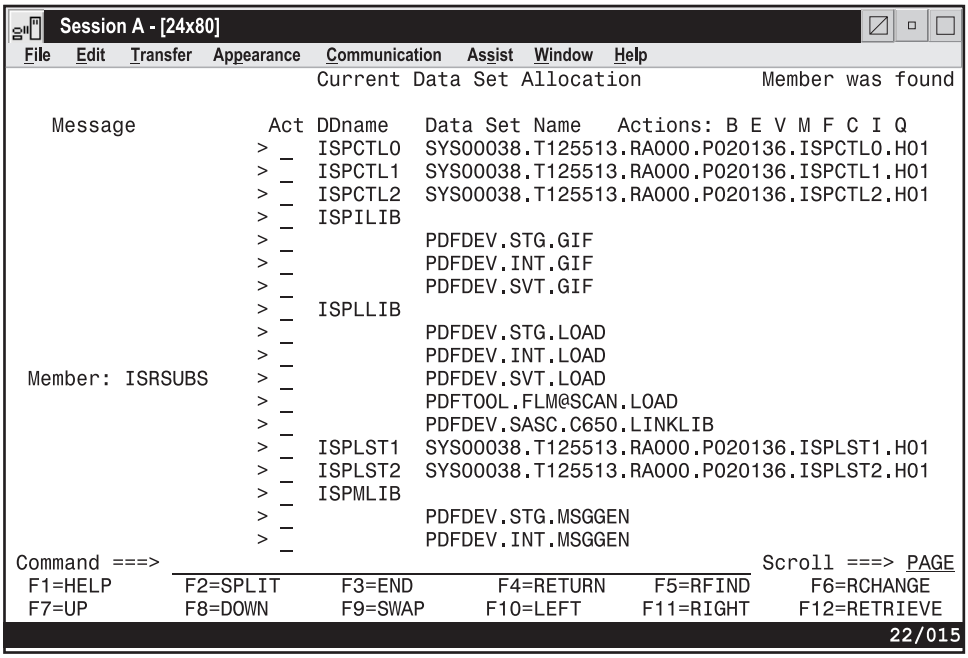


Figure 66. Results of the MEMBER Command

If the name is the name of a loaded module in the job pack area or LPA, you also see a panel similar to the one in Figure 69 on page 212.

When a member name is used on the MEMBER command (such as, M ISRSUBS) and an E, V, or B line command is used next to a data set in which that member is found, ONLY that member is Edited, Viewed, or Browsed. When the M line command is used, the member list is shown with the selected member at the top of the list.

When a member name pattern is used on the member command (such as, M ISR*), the E, V, B, and M line commands display member lists with members that match the given pattern.

Use the MEMBER command in situations when you do not know from where a member is coming or when you suspect that you might be accessing the wrong copy of a member. For example, if you are developing ISPF panels and you do not see your version of the panel being displayed, you can issue the MEMBER command to search for other copies of the panel.

Usually the MEMBER command operates on the entire displayed list. You can add a second operand that is a partial DD name. For example, the command **M ISRSUBS PL** searches only DD names with the string PL in it such as IS**PL**LIB or STE**PL**IB. This avoids having to use the ONLY command to limit the search.

Clist (CL) or Save (SA)

The CLIST command creates a CLIST that contains TSO ALLOCATE statements to reproduce the allocations in the displayed list. The CLIST name is 'userid.ISRDDN.CLIST' or 'prefix.userid.ISRDDN.CLIST'. You can use the command name SAVE instead of CLIST.

Use this command when you want to change allocations for testing purposes. For example, to add a panel library to your ISPPLIB concatenation:

- Enter ISRDDN
- Type **O ISPPLIB** to limit the displayed list to DD name ISPPLIB
- Type CLIST to create and edit the ISRDDN.CLIST data set
- Change the ALLOCATE statement to add your data set
- Exit ISPF
- Execute the CLIST (that is, EX ISRDDN)

Like the MEMBER command, you can add a whole or partial DD name to limit the number of DD names that are included in the generated CLIST. For example, to create a CLIST that only contains allocation statements for DD names containing the string ISP, type **CLIST ISP** or **SAVE ISP**.

Check (CH)

The CHECK command turns on or off automatic checking for mixed concatenations. CHECK or CHECK ON enables automatic checking, and CHECK OFF disables it. When checking for mixed concatenations is enabled, ISRDDN checks for concatenations with mixed record formats, mixed fixed record lengths, and mixed data set organizations. Because there are times when these concatenations are intended, you might want to turn off the warning generated by ISRDDN.

Count (C)

The COUNT command displays the number of members in a partitioned data set. The number of members is shown in the message area on the left side of the list.

COUNT can be used to find out if you have empty data sets in your concatenations. For example, if you want to find out if all members of an SCLM-controlled library system were successfully promoted, you can edit the hierarchy, invoke ISRDDN, and use the COUNT command to verify that all of the expected libraries in the concatenation are empty.

Like the MEMBER command, you can add a whole or partial DD name to limit the number of DD names that are searched.

Duplicates (DUP)

The DUPLICATES command searches all of the partitioned data sets in the displayed list and the LPA and displays a list of duplicate names. From the duplicates list, you can use the E (edit), B (browse), and V (view) line commands to view the PDS member or LPA storage. Use the DUPLICATES command to see where you might have potential conflicts with old or modified versions of load modules, REXX or CLIST programs, ISPF panels, or other PDS members.

For module names found in the Link Pack directory, the address of the module and its size are shown on the left side of the screen. If the name is an alias of a different module, the real name (major name) is shown instead of the size.

The duplicates list is shown in Figure 67 on page 211. Like the MEMBER command, you can add a whole or partial DD name to limit the number of DD names that

are searched. For example, to search only DD names that contain the string LLIB, enter **DUP LLIB**.

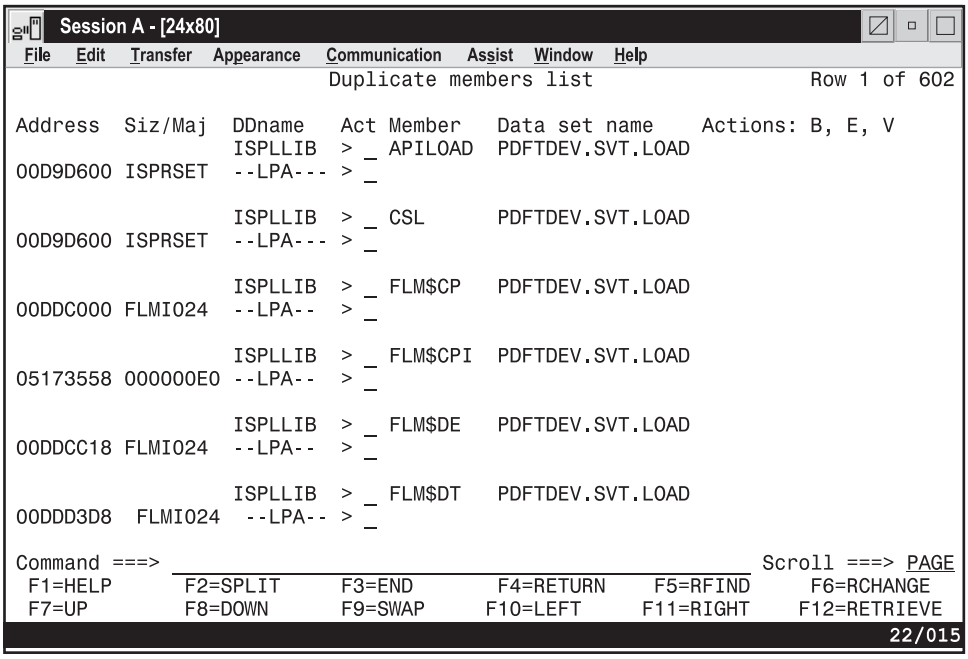


Figure 67. The Duplicates List Display

Apf, Linklist (LI), Parmlib, and Lpa (LP)

The APF, LINKLIST, PARMLIB, and LPA commands add and remove pseudo-DD names that show the defined APF libraries, link list libraries, PARMLIB libraries, and LPA libraries respectively. These pseudo-DD names are shown as if they are allocated DD names, but no actual allocation to the libraries is made. You can use most of the primary and line commands with these names, just as you would with real DD names.

In the confirmation panel, you can type the word **YES** to process the libraries, or **SKIP** to process the libraries and avoid the confirmation panel in the future. Dynamic LPA, Link lists, PARMLIB, and APF lists are all supported.

The LINKLIST and LPA commands add both the LINKLIST and LPALIB pseudo-DD names. To delete any pseudo-DD name, enter the appropriate command a second time. For example, to add APF libraries to the list, use the APF commmand. To remove the APF libraries from the list, enter the APF command a second time.

Select (S) and Load (L)

The SELECT command searches the job pack area (JPA) and link pack area (LPA) to see if a module is loaded. If the module is found, you see the CSVQUERY Results panel shown in Figure 69 on page 212.

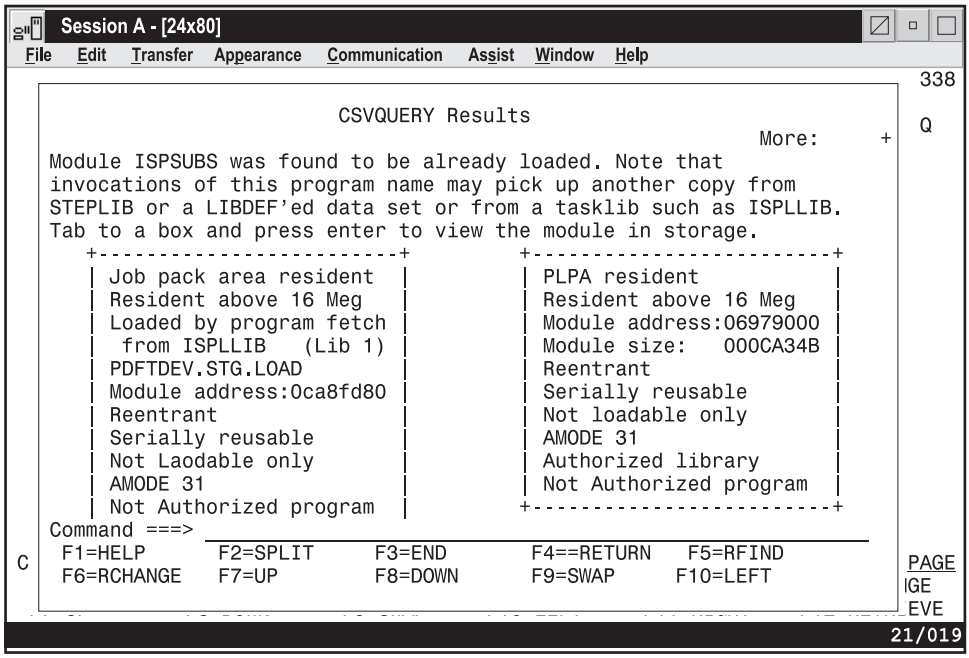


Figure 69. The CSVQUERY Results Panel

The information shown in the CSVQUERY Results panel is mostly derived by issuing a CSVQUERY macro. The data set name from which the module was loaded is shown if it can be determined. However, because of the way this information is gathered, the data set name can be incorrect if the original DD name from which the data set was loaded has been reallocated since the module was loaded.

On the CSVQUERY Results panel, you can use the TAB key to place the cursor inside the boxes describing the load module. If you then press Enter, you can browse the load module in storage.

The SELECT command is useful in situations where you need to know where a loaded program came from, for example, when you think you might be running mixed levels of ISPF or of an application under ISPF.

If a module is not loaded but you want to see its attibutes, you can use the LOAD command instead of the SELECT command. LOAD uses the current tasklib such as ISPLLIB, but you should verify that the loaded module came from the source you were expecting it to come from. LOAD automatically browses the load module storage.

Custom (CU)

The CUSTOM command shows several settings about your ISPF installation. It shows the values that used to be set in the ISPDFLTS CSECT but are now in the ISPF Configuration table, and it shows the values configurerd in module ISPTCM. This command is helpful when you are having trouble with the way certain programs are invoked. For more information about ISPTCM, refer to the *ISPF Planning and Customization Guide*.

MList (ML)

The MLIST command displays the eyecatchers for some of the ISPF CSECTs contained in modules ISPSUBS and ISRSUBS. This command can sometimes be used to verify that you are running with a particular level of maintenance because the eyecatchers in most ISPF modules contain a release number or a PTF level.

Browse (B)

ISRDDN provides a method of browsing storage using ISPF BROWSE. The storage can be browsed as unformatted data, as minimally formatted data, or as a side-by-side hexadecimal and EBCDIC dump format. ISRDDN also enables you to automatically chain lists, view arrays, and view the data pointed to by control blocks that are mainly lists of pointers (such as CVT).

The BROWSE primary command accepts a storage address, module name, or TSO TEST address locator string. Some examples invoking BROWSE are:

B ISRSUBS	Browse the already loaded module named ISRSUBS.
B 10.	Browse storage at hexadecimal location 10. In order to distinguish hexadecimal addresses from module names, absolute addresses must end with a period.
B 0.+21c?+b4?+108?+8	Browse storage based on a TSO TEST style string. In this case, the control block called the Protected Step Control Block or PSCB is shown.
B ISRSUBS+60?	Browse the address pointed to by the 4 bytes at offset hexadecimal 60 into module ISRSUBS.
B ? or B +0?	When executed from within the storage browser, this command uses the address 0 bytes from the beginning of the displayed storage as a pointer and starts a new browse session to show that storage.

Enq (E)

You can view ENQs on the system using the ENQ command. A display similar to the one shown in Figure 70 on page 214 appears. You can reduce the size of the list by specifying a QNAME, RNAME, address space name, and a system name. All entries are treated as prefixes, so you might not need to specify complete names.

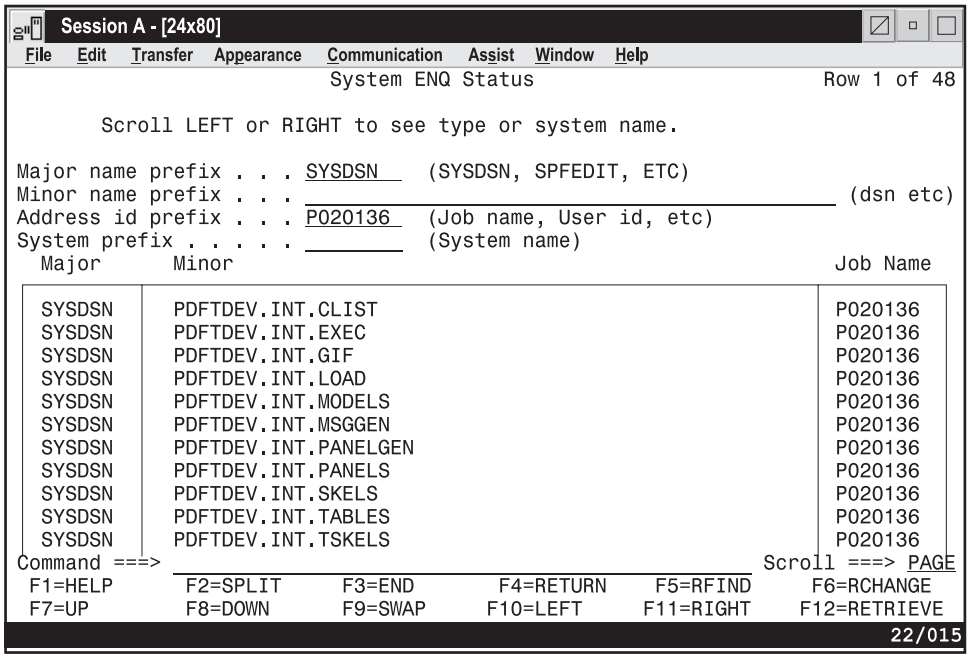


Figure 70. The System ENQ Status List Panel

The ***Major*** column shows the QNAME. The ***Minor*** field shows the RNAME and if the RNAME is 8 bytes or less, it shows the hexadecimal representation of the RNAME next to the EBCDIC representation.

In the System ENQ Status list, the ***job name*** field is color-coded to indicate the type of ENQ that the address space holds or is waiting for. Green indicates a shared ENQ. Red indicates an exclusive ENQ.

If an address space does not hold the ENQ but is waiting for it, the job name is shown highlighted in reverse video.

On narrow screens, you can scroll right or left for more information. By scrolling left and right you see the system name and ENQ options (SYS (system), SYSS (systems), STEP, G (global), and R (reserve)). On wide screens you see all of the information on one screen without scrolling left or right.

On the System ENQ Status display, press END to return to the Current Data Set Allocations list, or enter CON to view the System ENQ Contention display. You can also use the ALL command to view all ENQs or use the RESET command to see only the data set ENQs (QNAME SYSDSN) for your TSO user id.

Con (C)

You can view ENQ contention on the system by using the CON command. When ENQ contention exists, you see a screen similar to the one in Figure 70, but without the input fields. When no contention exists, a message displays instead of the list.

Allocation List Line Commands

Allocation list line commands are entered next to a DD name or data set. By default the allocation list is in ***short*** format. This means that for concatenations, the DD name is next to the first data set name in the concatenation.

When a line command is entered next to a DD name, the command is intended to work on the DD allocation rather than the data set name on that line. For example, an **E** command next to the DD name that refers to a concatenation edits the whole concatenation. If you want to edit just the first data set in a concatenation, use the LONG command to place the list in ***long*** format. In long format, the DD name for a concatenation is on a separate line so that you can place line commands next to the first data set name in the concatenation.

The Edit, Browse, View, and Member list commands are sensitive to the results of the MEMBER primary command. When the MEMBER primary command searches the displayed list for a member or members matching a name pattern, the member or pattern is shown in the list. Placing an E, B, V, or M next to a name in which the member or pattern was found displays either a member list with member names matching the pattern or the specific found member.

E - Edit

The E line command edits a data set or concatenation. It can be used on any data set or any DD name allocated to a data set (real or VIO). You might want to use the E line command for editing temporary files such as JCL that was created by file tailoring and written to the ISPCTLn DD name.

B - Browse

The B line command browses a data set or concatenation. It can be used on any data set or any DD name allocated to a data set (real or VIO). You can use the B line command for browsing allocated files. For example, the compress option in the PDF utilities, option 3.1, creates a listing data set that is sometimes allocated to the ISPCTL1. When you press the HELP key after compressing a data set in option 3.1, you might see that the listing was saved in a temporary data set. The B line command in ISRDDN is an easy way to browse that data set.

V - View

Use the V line command to view a data set or concatenation. This is similar to E (Edit) but there is no SAVE command. Use this when you want to view a data set and modify it for easier viewing without risking changes to the data set.

M - Member list

The M command displays an enhanced member list for a data set or concatenation. This gives you greater flexibility in working with allocated data sets. You might use this command when you have several different operations to perform on members.

F - Free

Use the FREE command to free an allocation. The Free command must be specified next to a DD name. F commands next to data sets in a concatenation with an F next to the DD name are ignored because those data sets are removed from the list before the F commands are processed.

The F command uses SVC 99 (dynamic allocation) to free the DD name. However, if SVC 99 cannot free the data set, ISRDDN invokes TSO's FREE command. The TSO FREE command might write a message to the screen with information on why the free failed. This command is useful when you need to free allocations such as those left by prematurely terminated or poorly behaved programs.

C or Z - Compress

Use the COMPRESS command to compress partitioned data sets. The COMPRESS command can be used with data sets that are allocated as shared and can be used next to data set names or DD names.

I - Information

The I command attempts to invoke the PDF data set information utility to display information about a data set. It can be used next to any real data set name. VIO data sets are not supported. This command can provide information such as the number of allocated directory blocks or a data set's SMS management class, or other information that is not shown by scrolling the Current Data Set Allocations list left or right.

Q - Query ENQs

The Q command shows all SYSDSN and SPFEDIT ENQs that exist for a data set. This command is useful when you want to see what other users or jobs are using a data set you have allocated. Using the Q command provides the same information as using the ENQ primary command and selecting an RNAME of the data set name.

T - Test Directory

The T line command reads the directory of a PDS directly, and performs a BLDL command on each member to see if the BLDL service returns accurate information for the directory. The results are displayed in a separate Browse session. The T command can be used to debug problems such as I/O errors or the need to refresh LLA or other directory caching systems.

K- VTOC Information

The K line command displays VTOC information for the first volume on which the data set resides. The information reutrned is the same as you would see in the Data Set List utility (option 3.4, command V). You can also view VTOC information by placing the cursor on the volume name and pressing the Enter key. If the data set is not on a physical volume, the K command does not provide any information.

Browsing Storage

By using the BROWSE command within ISRDDN, you can view the contents of storage within your address space. When you are browsing storage, you can use any of the standard ISPF Browse primary commands. In addition, there are several primary commands you can use to format and move around in the storage list.

If you are not using one of the special display formats (CHAIN, ARRAY, or ARRAYP), you can scroll UP even when the ″Top of Data″ line is displayed so that you can see what data exists before your requested storage location. After you scroll up once, you can scroll up or down to the limits of the contiguous addressable storage.

BROWSE

When the BROWSE command is invoked within an existing browse session, a new browse session is started. The END command returns you to the previous Browse session. To end all of the sessions and return to the Current Data Set Allocations list, use the CANCEL command.

Browse can also be used with point and shoot. Place the cursor over an address within the display and press Enter. A new browse session is started to view the storage pointed to by the cursor. If the cursor is not on a valid, accessible address, an error message displays.

RAW

Display storage data as unformatted text. Storage is shown on the screen without any formatting. The data on a line is the data that immediately follows the data on the previous line. Because the FIND command is actually searching the screen image and not storage itself, it is best to search storage while in the RAW display format. Note that even in unformatted displays, if your search string would span lines, FIND does not locate the string. To avoid this, search for the string in RAW format, then enter the command **B +20** to find the string again. This shifts the display by 32 bytes (hexadecimal 20) and the line breaks occur in different places.

DATA

Display the storage as an unformatted string of data with offsets shown on the left side of the screen. Use this format to give you a better context of the storage you are browsing.

FORMAT

Display the storage in both hexadecimal and EBCDIC, the way you might see the format in a data dump (see Figure 71 on page 218). On wide screens such as a 3278 Mod 5, the format includes 16 bytes per line (8 sets of 4 bytes). On narrower screens, the format contains 8 bytes per line (4 sets of 4 bytes). FORMAT is the default format that appears whenever a storage browse session is started.

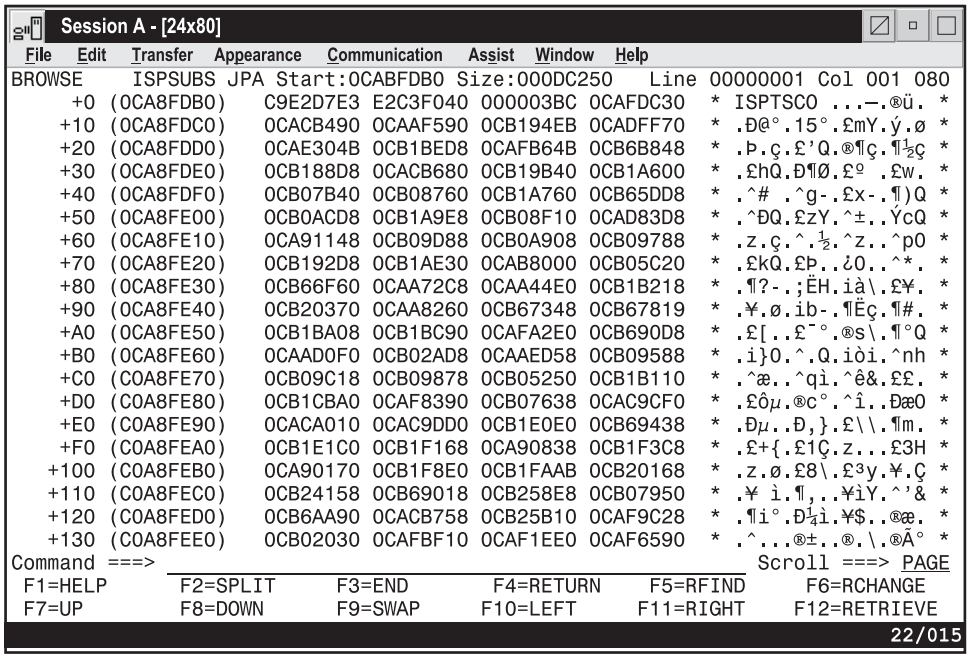


Figure 71. Storage Viewed in FORMAT Mode

WIDE and NARROW

The WIDE and NARROW commands switch the display between wide and narrow formats regardless of the type of terminal you are using. If the screen is too narrow to handle the wide format, you must scroll right and left to see all of the data.

REFRESH (REF)

Use the REFRESH command to scroll the display back to the +0 offset. REFRESH is useful if you have scrolled up past the initial "Top of Data" line and want to return to your original referenced storage location.

REFRESH is not available in CHAIN, ARRAY, or ARRAYP formatted displays.

CHAIN

When you are viewing a linked list, you can use the CHAIN command to view more than one link at a time. The syntax for CHAIN is

```
CHAIN [offset [length ]]
```

The *offset* is a hexadecimal offset of the 4–byte pointer to the next link. The *length* is a hexadecimal minimum length to show for each link. The default for offset is zero. The default for length is whatever fits on one line in the current display format. The chain is considered terminated when a pointer of zero is found, a pointer to the first node is found, or a pointer to unavailable storage is found. Entering the CHAIN command a second time turns the CHAIN formatting off.

For example, to see the current ASCB chain, enter ISRDDN and type

```
B 10.??+C?
CHAIN 4 20
```

You see a screen similar to Figure 72.

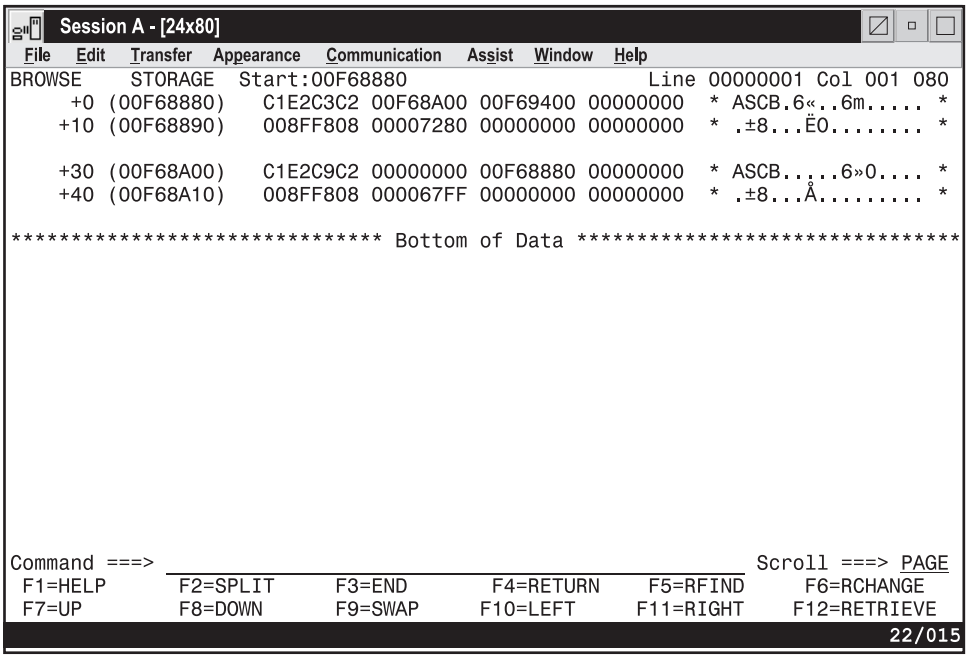


Figure 72. CHAIN Storage Format

In the CHAIN format display, the offsets on the left are the offsets within a particular link followed by the actual address of the start of the line.

ARRAY

When you are viewing that is an array, you can show the array elements as separate blocks of storage. The syntax for ARRAY is

```
ARRAY [dimension [length ]]
```

The ***dimension*** is the number of array elements in decimal. The ***length*** is the length of each element in hexadecimal. The default for length is 4.

For example, the static link list table is an array. Assuming that each element is 45 bytes (hexadecimal 2D) and that you want the first 30 entries, enter ISRDDN and type

```
B 10.?+4DC?+8
ARRAY 30 2D
```

You see a screen similar to Figure 73 on page 220.

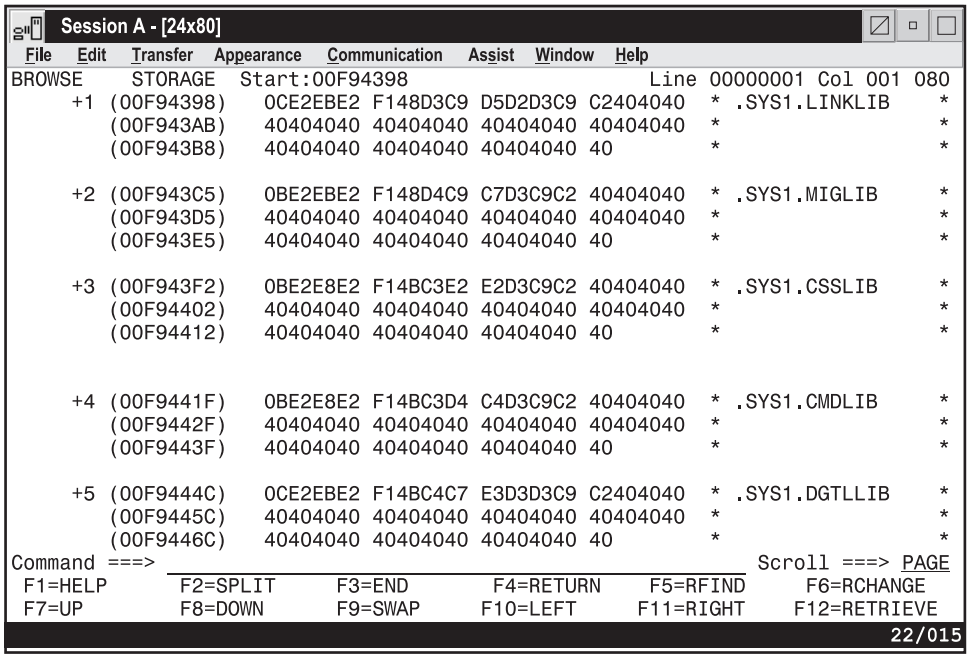


Figure 73. ARRAY Storage Format

In the ARRAY format display, the offsets on the left are the array element number followed by the address of the displayed line.

ARRAYP

Many control blocks are mainly list of pointers. For example, the Communications Vector Table (CVT) is an OS/390 control block that points to many other control blocks. The ARRAYP command shows the data pointed to by the pointers in a control block. Use the ARRAYP command when you are looking for the offset of a pointer to a particular storage location. The syntax for ARRAYP is

```
ARRAYP [dim [length ]]
```

The ***dim*** is the decimal number of pointers in the array. The ***length*** is a hexadecimal length of each element pointed to by the pointer. The default for length is whatever fits on one line in the current display format.

For example, to see what is pointed to by the elements of CVT, enter ISRDDN and type

```
B 10.?
ARRAYP
```

You see a screen similar to Figure 74 on page 221.

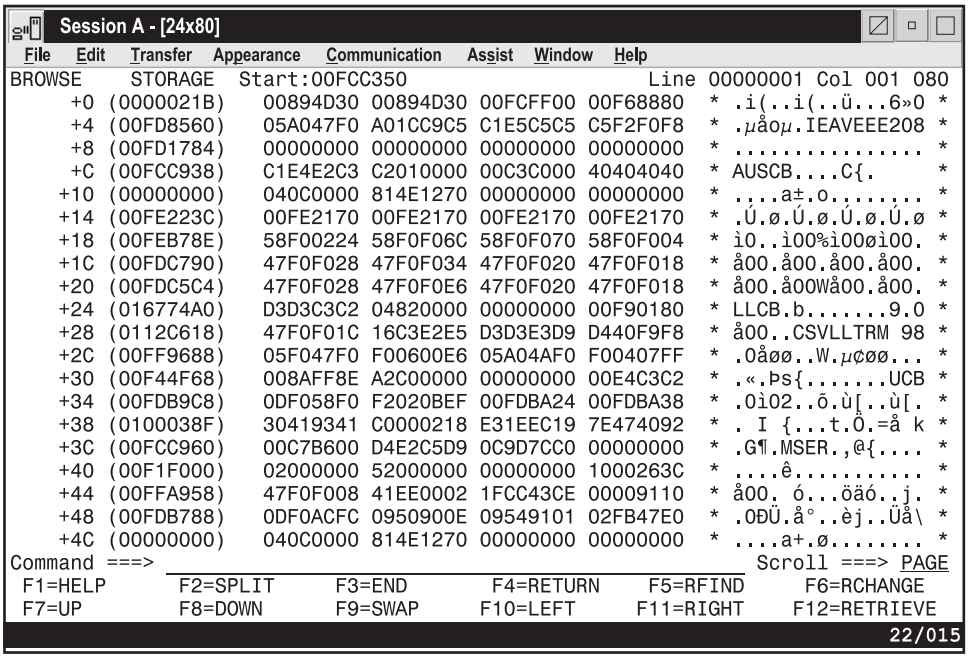


Figure 74. ARRAYP Storage Format

In the ARRAYP format display, the offsets on the left are the offsets within the array of pointers followed by the pointer itself. This is followed by the data to which the pointer refers.

Defining Named Storage Locations

If you browse the same storage locations or control blocks frequently, you might want to set up a file that names those storage locations so that you can use a name in the BROWSE command.

To enable the BROWSE command to use a named reference to storage, you must allocate a sequential file to the DD name ISRDDN. Within that file each line is either a comment or a named storage location. Comments start with a semi-colon (;).

Location definitions have a name as the first word, followed by a TSO TEST style locator string. Anything after the locator string is ignored. The TSO TEST locator string can use another defined name as a starting point. If the locator string cannot be resolved because of syntax or other errors, the line is ignored.

For example, if you allocate a sequential file like the one shown in Figure 75 to DD name ISRDDN, you could then browse your User Profile Table, which stores your TSO PROFILE settings, by typing **B UPT** on the command line.

```
CVT         10.?          Communications Vector Table
PSCB        JSCB+108?     TSO Protected Step Control Block
JSCB        TCB+B4?       Job/Step Control Block
TCB         CVT??         Task Control Block
UPT         PSCB+34?      User Profile Table
```

Figure 75. Sample ISRDDN Named Storage File

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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504–1785, USA.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries in writing to

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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 OR 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 publication 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 the IBM Corporation, Department TL3B, 3039 Cornwallis Road, Research Triangle Park, North Carolina, 27709–2195, USA. 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 document 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.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This book primarily documents information that is NOT intended to be used as Programming Interfaces of ISPF.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

BookManager
C++
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSMS/MVS
DFSORT
ESCON
FFST
GDDM
IBM
Language Environment
MVS
MVS/ESA
OS/2
OS/390
OS/390 Security Server
RACF
Resource Access Control Facility
SOMobjects
System View
VisualLift
VTAM

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Z

Readers' Comments — We'd Like to Hear from You

Interactive System Productivity Facility (ISPF)
User's Guide Volume I
z/OS Version 1 Release 2.0

Publication No. SC34-4822-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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.

Name

Address

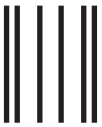
Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC34-4822-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Software Reengineering
Department G7IA / Bldg 503
Research Triangle Park, NC
27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

SC34-4822-01

Cut or Fold
Along Line

File Number: S370/4300-39
Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC34-4822-01