z/OS

Resource Measurement Facility Programmer's Guide

SC33-7994-03

z/OS

Resource Measurement Facility Programmer’s Guide

SC33-7994-03

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page C-1.

Third Edition, July 2003

This edition applies to Version 1 Release 4 of z/OS (5694-A01), z/OS.e Version 1 Release 4 (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

```
IBM DEUTSCHLAND ENTWICKLUNG GMBH
eSERVER PERFORMANCE MANAGEMENT DEVELOPMENT
SCHOENAICHER STRASSE 220
71032 BOEBLINGEN
GERMANY
```

If you prefer to send comments electronically, use one of the following methods:
FAX (RMF Development): Your International Access Code +49+7031+16+4240
Internet: rmf@de.ibm.com

Internet

Visit our homepage at http://www.ibm.com/servers/eserver/zseries/zos/rmf/

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

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

Contents

Figures

Tables

About this document

The Resource Measurement Facility (RMF) is the element of z/OS and z/OS.e for performance management.

This document contains information and reference material to enable you to use RMF data for application programming. There is a number of different ways of getting at different kinds of information, and each one is described in a separate chapter of this document.

Further processing of RMF report data can also be done using spreadsheets. The Spreadsheet Reporter is described in the *z/OS RMF User's Guide*.

In addition, this document is describing diagnosis procedures that can be used in case of an error when running RMF.

Who should use this document

This document is intended for use by system programmers responsible for the development of individual, installation-specific applications in the area of system measurement. Because RMF is a tool for measuring MVS system performance, this document assumes that the reader has extensive knowledge of the MVS system.

For an overview of RMF, and guidance on using the standard capabilities of the product, see the *z/OS RMF User's Guide*.

How this document is organized

This document contains the following chapters:

Chapter 1, "SMF records"
These are the records from which RMF obtains information for the standard reports. You can find all the information you need to use them for your own reports in this chapter.

Chapter 2, "RMF sysplex data services"
These are callable services with which you as an RMF user can access performance data sysplex-wide. The calls, return codes and data layouts are described here.

Chapter 3, "Using LDAP to access performance data"
Using LDAP, you can access RMF performance data from application programs. The information contained in this chapter describes how RMF data is stored within LDAP and how to access it from C and Java applications as well as from a web browser which supports the LDAP protocol.

Chapter 4, "Adding Monitor I and Monitor II installation exits"
You can enhance the gathering capabilities of Monitor I and add your own report types to Monitor II by writing your own exit routines. Details on coding and installing these exit routines are given in this chapter.

Chapter 5, "Adding Monitor III user exits"
The RMF Monitor III Utility helps you to add your own processing to the standard Monitor III reporting. This chapter describes this utility and its usage.

Chapter 6, "Using Monitor III VSAM data set support"
The processing and format of the VSAM data sets that Monitor III uses to store its information are described in this chapter.

Chapter 7, "Monitor III data reporter tables"
When coding Monitor III exit routines, for example, with the help of the Monitor III Utility, you have to know what information RMF has stored where for use in which reports. The data is stored in tables, and the layouts of these are shown here.

Chapter 8, "Diagnosis reference"
In this chapter, you find procedures that you might use in case of an error when running RMF.

The z/OS RMF library

This table shows the full titles and order numbers of the books in the RMF library for z/OS.

Table 1. RMF Library

Title	**Order Number**
Books available as Hardcopy and Softcopy	
z/OS *z/OS RMF User's Guide*	SC33-7990
z/OS *z/OS RMF Report Analysis*	SC33-7991
z/OS *z/OS RMF Performance Management Guide*	SC33-7992
z/OS *z/OS RMF Messages and Codes*	SC33-7993
z/OS *z/OS RMF Programmer's Guide*	SC33-7994
z/OS *z/OS RMF Reference Summary*	SX33-9033
Softcopy documentation as part of the *z/OS Collection* (SK3T-4269 (CD-Rom) and SK3T-4271 (DVD))	
z/OS *z/OS RMF NewsFLASH*	SC33-7995

Related information

For additional information on z/OS, see the *z/OS Information Roadmap*, SA22-7500.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages and system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

You can use LookAt on the Internet at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html`

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on the **News and Help** link or from the *z/OS Collection*, SK3T-4269.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id* as in the following:

```
lookat erb100i
```

This results in direct access to the message explanation for message ERB100I.

To find a message explanation from the LookAt Web site, simply enter the message ID and select the release you are working with.

Summary of changes

What's new in z/OS V1R4

Summary of Changes
for SC33-7994-02
z/OS Version 1 Release 4 (z9xx Compatibility)

This document contains information previously presented in *z/OS RMF Programmer's Guide*, SC33-7994-01, which supports the Resource Measurement Facility.

This document includes terminology, maintenance, and editorial changes. The following information describes the enhancements that are being distributed with z/OS Version 1 Release 4. All technical changes or additions to the text are indicated by a vertical line to the left of the change.

> **Note:**
> An appendix with z/OS product accessibility information has been added.

New Information

Monitor III VSAM Data Set Support

Table ERBCFIG3 describes the structure of records containing coupling facility information.

Table ERBCATG3 describes the structure of records containing cache data information.

Changed Information

Monitor III Data Reporter Tables

Table ERBCDCT3 for the Monitor III **CPC Capacity** report has been renamed to ERBCPCT3 and entries have been deleted from this table.

RMF Phase Driver Table ERBPHDS3 has been updated.

Monitor III VSAM Data Set Support

The following tables have been updated:

- ERBRCDG3
- ERBASIG3
- ERBCPUG3
- ERBDVTG3
- ERBGEIG3
- ERBUWDG3

LDAP Directory Structure

Object class `rmfCPC` has been added to the LDAP directory structure.

History of changes

What's new in z/OS V1R2

New Information

SMF Records
There is a new subtype 2 for SMF record type 70 with activity data of cryptographic processors. Data of the traditional record type 70 is described as type 70 subtype 1.

Using LDAP to Access Performance Data
Using LDAP, you can access RMF performance data from application programs. This allows other performance and systems management components easy access to performance data within an z/OS environment using standard LDAP clients, for example, Java and C programs or a web browser which supports the LDAP protocol.

Monitor III Data Reporter Tables
New fields for the support of HiperSockets have been inserted into table ERBCHAT3.

Table ERBCDCT3 is available for the new Monitor III **CPC Capacity** report which provides information about defined and consumed processor capacity of all partitions in a CPC (central processor complex). This report is available as SPE and needs to be installed as APAR OW49807.

The field IOQACBVC has been removed from table ERBIOQT3.

Changed Information

Monitor I and II Installation Exits
There is no longer a local 3270 display session for Monitor II. However, you can have access to Monitor II reports without an active TSO/TCAS subsystem by means of the RMF Client/Server Enabling (RMFCS).

Monitor III VSAM Data Set Support
Tables ERBASIG3 and ERBRCDG3 have been updated.

What's new in z/OS V1R1

The following information describes the enhancements that are being distributed with z/OS Version 1 Release 2. It is indicated by a vertical line to the left of the text.

There is one release RMF 2.10 that can run in OS/390 2.10 as well as in z/OS V1R1. Nevertheless, some new functions are available only when z/OS is running on a zSeries 900 server.

New Information

SMF Records
There is a new SMF record type 74-7 for FICON director activities. Record types 78-1 and 79-13 for measurement data of 4381 processors are no more available.

Diagnosis Reference
The contents of the *RMF Diagnosis Guide* has been integrated into this publication.

Changed Information

z/OS Version 1 Release 1

Monitor III VSAM Data Set Support
Table ERBASIG3 has been updated.

Spreadsheet Converter
The description of the Spreadsheet Converter has been removed. The Spreadsheet Converter is part of the Spreadsheet Reporter, and it is described in the *z/OS RMF User's Guide*.

z/OS Version 1 Release 1

Chapter 1. SMF records

SMF Records Written by RMF

This chapter covers the following items:

- Summary of all RMF/SMF record types
- How to archive and print SMF records
- How to obtain SMF records directly

Overview

Each SMF record contains information similar to the contents of the corresponding formatted report. For each system activity that you select, RMF collects data and formats an SMF record to hold the data it collects.

Some totals, averages, and percentages are not explicitly contained in the SMF records, but are calculated from the SMF data. For elaboration of particular fields, see the descriptions of the corresponding fields in the printed report descriptions in *z/OS RMF Report Analysis*.

Also, each SMF record produced by RMF is described in *z/OS MVS System Management Facilities (SMF)*.

RMF does not generate reports from SMF records type 72, subtypes 2 or 4. However, these records are available for user-written reports.

Define the SMF record types and subtypes to be written in the SMFBUF option, which you can specify:

- In the PARM field of the RMF cataloged procedure
- On the system command START RMF
- On the system command MODIFY RMF

The record types and the corresponding RMF measurement activities are:

- Record type 70 has the following subtypes:
  - Subtype 1 – CPU and PR/SM activity
  - Subtype 2 – Cryptographic processor activity
- Record Type 71 – Paging activity
- Record type 72 has the following subtypes:
  - Subtype 1 – Workload activity (compatibility mode)
  - Subtype 2 – Storage data (compatibility mode)
  - Subtype 3 – Workload activity (goal mode)
  - Subtype 4 – Storage data (goal mode)
- Record Type 73 – Channel path activity
- Record type 74 has the following subtypes:
  - Subtype 1 – Device activity
  - Subtype 2 – XCF activity
  - Subtype 3 – OMVS Kernel activity
  - Subtype 4 – Coupling facility activity
  - Subtype 5 – Cache subsystem activity
  - Subtype 6 – Hierarchical file systems statistics
  - Subtype 7 – FICON director statistics
- Record Type 75 – Page/Swap data set activity
- Record Type 76 – Trace activity
- Record Type 77 – Enqueue activity
- Record type 78 has the following subtypes:
  - Subtype 2 – Virtual storage activity
  - Subtype 3 – I/O queuing activity
- Record type 79 has the following subtypes for Monitor II snapshot data:
  - Subtype 1 – Address space state data
  - Subtype 2 – Address space resource data
  - Subtype 3 – Central storage/processor/SRM
  - Subtype 4 – Paging
  - Subtype 5 – Address space SRM data
  - Subtype 6 – Reserve data

- Subtype 7 – Enqueue contention data
- Subtype 8 – Transaction activity data
- Subtype 9 – Device activity
- Subtype 10 – Domain activity
- Subtype 11 – Paging activity
- Subtype 12 – Channel path activity
- Subtype 14 – I/O queuing activity
- Subtype 15 – IRLM long lock detection

You find details about which monitor is writing what SMF records in the *z/OS RMF User's Guide*.

SMF record format

Programming Interface information

Depending on the feedback options you select, RMF can write the SMF records to the SMF data set, use the data in the record to generate a printed report, or both. Regardless of the options you select, the format of the SMF record is the same.

Each SMF record that RMF generates consists of the following sections:

1. **SMF common header**, which identifies the record length, the record type, the time and date, the SMF system identifier, the subsystem identifier (always RMF), and the record subtype (if required). It also describes the other sections in the record. Each section is identified by its offset, the length of the section, and the number of such sections in the record. These offset/length/number triplet pointers define the structure of the rest of the record.
2. **RMF product section**, which includes information such as the RMF version number, the start time of the interval, the length of the interval, the length of the sampling cycle, and interval synchronization data. The RMF product section is the same in all records.
3. **Control section**, which contains general one-time data for RMF to use to produce any requested report. The contents of the section depend on the record type. Some records do not require a control section, while others require more than one.
4. **Data section**, which includes the specific data gathered during the interval. The format and the number of the data sections depend on the record type and the data collected. For example, there would be one data section for each device included in the type 74 record, I/O device activity.

With this format, the SMF records that RMF generates can change to incorporate any new or modified data without creating incompatibilities. The key factors in allowing for compatible change are the grouping of similar data in one section and the use of the offset/length/number triplet pointers to access the data stored in each section. Figure 1-1 shows the general format of the SMF records that RMF generates. The figure shows both the pointer structure and the storage layout for the sections.

Also, you can access fields in the SMF common header and the RMF product section by either a general name or a specific name. For example, you can access the interval start time in a type 70 record by either its general name (SMFIST) or its specific name (SMF70IST). Thus, code that processes all records can use the general name while code that processes only a specific record type can use the specific name.

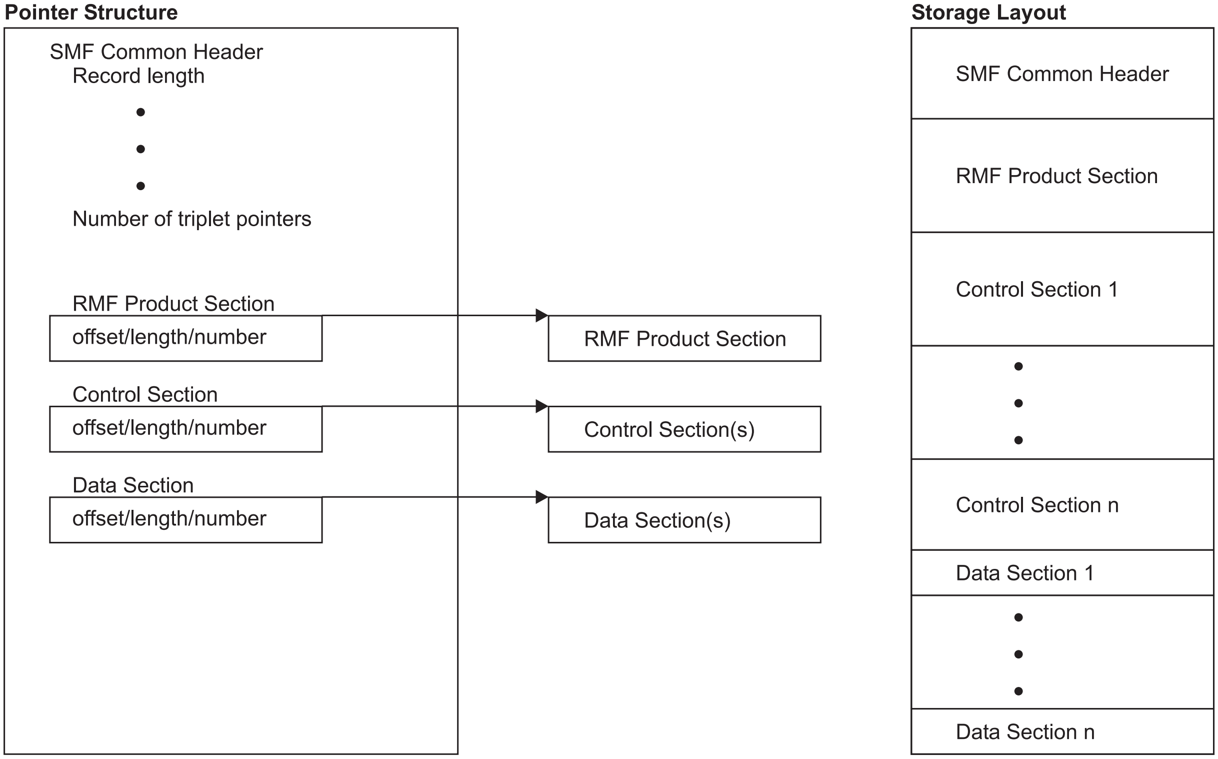


Figure 1-1. SMF Record Format

If your installation has existing data reduction programs that use SMF record input, check the SMF record formats carefully to determine what changes are required. Note that using the SMF record mapping macro instructions supplied by RMF is the most flexible way to access the contents of the SMF records your programs require. When you use the mapping macros, usually only a re-assembly of your program is required to incorporate changes to the record format.

The SMF record mapping macro instruction is ERBSMFR. Its format is:

```
ERBSMFR(nn1[,nn....1])
```

where nn identifies the type(s) of the SMF record(s) you want to map. Note that the parentheses are required only when two or more SMF record types are specified.

If you specify ERBSMF, the macro generates a mapping of the SMF common header and the RMF product section using only the general names.

The mapping macros reside in SYS1.MACLIB.

Because RMF can generate spanned SMF records – particularly when I/O device activity is measured – correct DCB parameters are important. Do not override the DCB parameters in the data set label by specifying DCB parameters on JCL statements. However, when using unlabeled tape the JCL describing an input SMF record data set should specify RECFM=VBS and a logical record length (LRECL) that is at least equal to the length of the longest record.

End of Programming Interface information

Archived performance data

You may find it useful to archive the performance data collected in the SMF records RMF produces. You can use this data to study trends or to evaluate the impact of a system change. Because of system changes and/or RMF changes, the archived data recorded by various versions or releases of RMF is not always the same. The SMF record level change number field in all RMF SMF records lets you process any SMF record changes that may result from later RMF releases.

RMF version numbers

Programming Interface information

The Postprocessor reads the RMF version number of each SMF record in the input stream. This number appears in field name SMFxxMFV, where xx is the record number, the field contains one of the following values:

- X'606F' for an SMF record produced by OS/390 2.6.0 RMF
- X'607F' for an SMF record produced by OS/390 2.7.0 RMF
- X'610F' for an SMF record produced by OS/390 2.10.0 RMF

  This version number is also correct if RMF is running under z/OS V1R1.
- X'712F' for an SMF record produced by z/OS V1R2 RMF

When the version number indicates that the record was produced by an earlier version or release of RMF, the Postprocessor converts the record to the current RMF format. A converted record, however, is not exactly the same as a current record. The major differences are:

- Fields for data that only the current version of RMF collects contain blanks or zeroes in the converted record.
- Fields for data that will not be collected anymore are omitted.
- The converted record contains a flag that indicates that it is a converted record, but RMF does preserve the original record version number.

These differences will also be reflected accordingly in the reports.

End of Programming Interface information

Printing SMF records

You might occasionally find it necessary to print the SMF records RMF produces. Printed records are useful, for example, when designing and implementing a user-written record processing program or when diagnosing problems with RMF reports. There are two ways to print the records:

- The standard utility program IDCAMS - it can print all SMF records in dump format.
- The RMF utility program ERBSCAN running under ISPF - it can format all SMF/RMF records in record-type-specific sections.

Using the IDCAMS utility

A sample of the JCL needed to print SMF records follows. The first step (SELECT) limits the amount of output to the record types or time frames that you need. If you want to print the entire data set, use only the second step (PRINT), defining there the data set with the SMF records. These JCL statements and SMF dump parameters select and print SMF record type 74 that were written from 19:00 AM until 19:15 AM on April 3, 2002.

```
//SELECT    EXEC  PGM=IFASMFDP
//SYSPRINT  DD    SYSOUT=A
//IN        DD    DSN=data set containing SMF records
//OUT       DD    DSN=&&RMFREC,DISP=(NEW,PASS),UNIT=SYSDA
//SYSIN     DD    *
  INDD(IN,OPTIONS(DUMP))
  OUTDD(OUT,TYPE(74))
  START(1900)
  END(1915)
  DATE(20020403,20020403)
/*
//PRINT     EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=A
//RMFREC    DD    DSN=&&RMFREC,DISP=(OLD,PASS)
//SYSIN     DD    *
  PRINT   INFILE(RMFREC)
/*
```

z/OS MVS System Management Facilities (SMF) contains more information on the IFASMFDP dump program. *z/OS DFSMS Access Method Services for Catalogs* contains more information about IDCAMS.

Because you do not specify the format on the PRINT statement, the format defaults to DUMP. The records are printed in a dump format. Figure 1-2 is an example of the SMF record dump format. The offsets are in the left column, and the right side of the dump contains a printable section to help find the fields of interest. Note that the PRINT utility does not include the record length and segment descriptor fields in its output. As a result, a field shown at offset 4 in an SMF record in *z/OS MVS System Management Facilities (SMF)* appears at offset 0 in the formatted dump. You must adjust subsequent offsets accordingly to refer back and forth from the formatted dump to the printed SMF records in *z/OS MVS System Management Facilities (SMF)*.

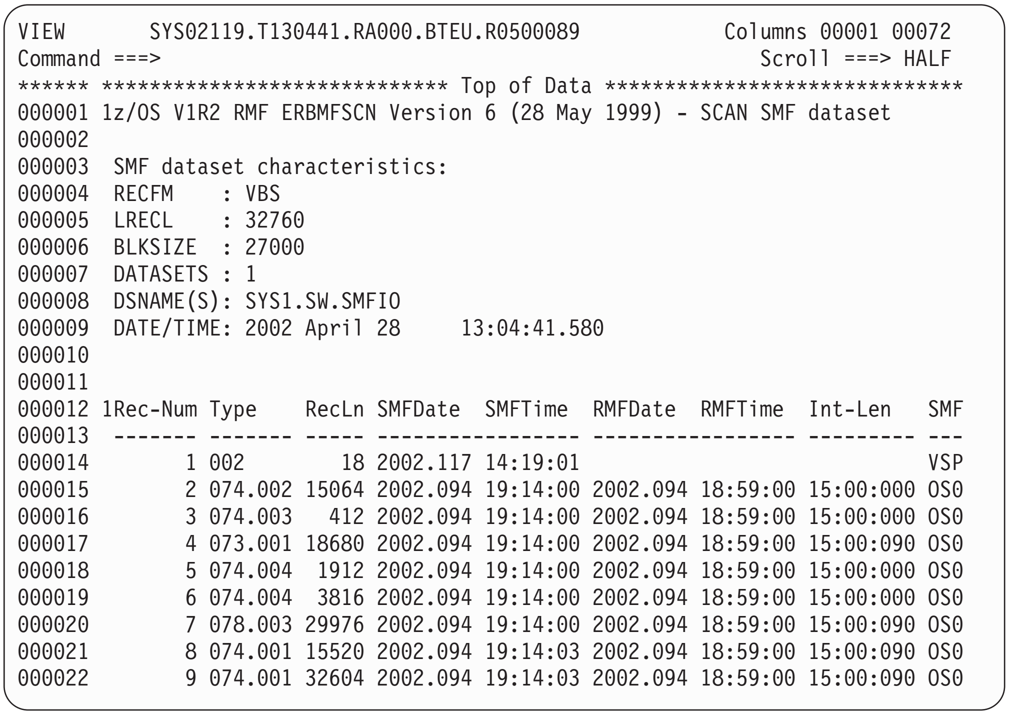
```
IDCAMS  SYSTEM SERVICES                                           TIME: 13:28:15       04/28/02    PAGE      2

LISTING OF DATA SET -SYS1.SW.SMFIO

RECORD SEQUENCE NUMBER - 1
000000  1E02004E A56D0100 117FE5E2 D7D4                                             *...+._..."VSPM                *

RECORD SEQUENCE NUMBER - 2
000000  DF4A0069 A70B0100 094FD6E2 F0F4D9D4   C6400002 00050000 00000044 00680001   *.........|OS04RMF ..............*
000020  000000AC 001C0001 000000C8 0038000E   000003D8 00580040 000019D8 002C00C0   *...........H.......Q... ...Q...{*
000040  606FD9D4 C6404040 40400185 900F0100   094F1500 000F0000 0000005A 00003000   *-?RMF     .......|.........!....*
000060  40404040 0009999F E5C5F0F2 F0F6F0F0   03E00438 B3D6ECD1 1A600000 00001AD2   *    ....VE020600.\...O.J.-.....K*
000080  74800000 00000000 00000000 03840DD4   B3D6ECD1 1A600000 E2D7D3C5 E7F0F140   *...............M.O.J.-..SPLEX01 *
0000A0  D6E2F0F4 40404040 00000000 00000000   00000000 00000000 00000000 00000000   *OS04    ........................*
0000C0  00000000 D6E2F0F1 40404040 00800000   00000008 00000000 00000000 00001B58   *....OS01    ....................*
0000E0  00000000 00000000 00000000 00000000   00000000 40404040 40404040 D6E2F0F1   *....................        OS01*
000100  40404040 00400000 00000002 00000000   00000000 00000ABE 00000000 00000000   *    . ..........................*
000120  0000007A 00000000 00003FBC C4C5C6C1   E4D3E340 D6E2F0F1 40404040 00400000   *...:........DEFAULT OS01    . ..*
000140  00000006 00000000 00000000 00001676   00000000 00006889 00000000 00000000   *................................*
```

Figure 1-2. Dump Format of SMF Record

Using the ERBSCAN utility

You can use the ERBSCAN utility to display RMF records directly under ISPF.

ERBSCAN *data-set-name*



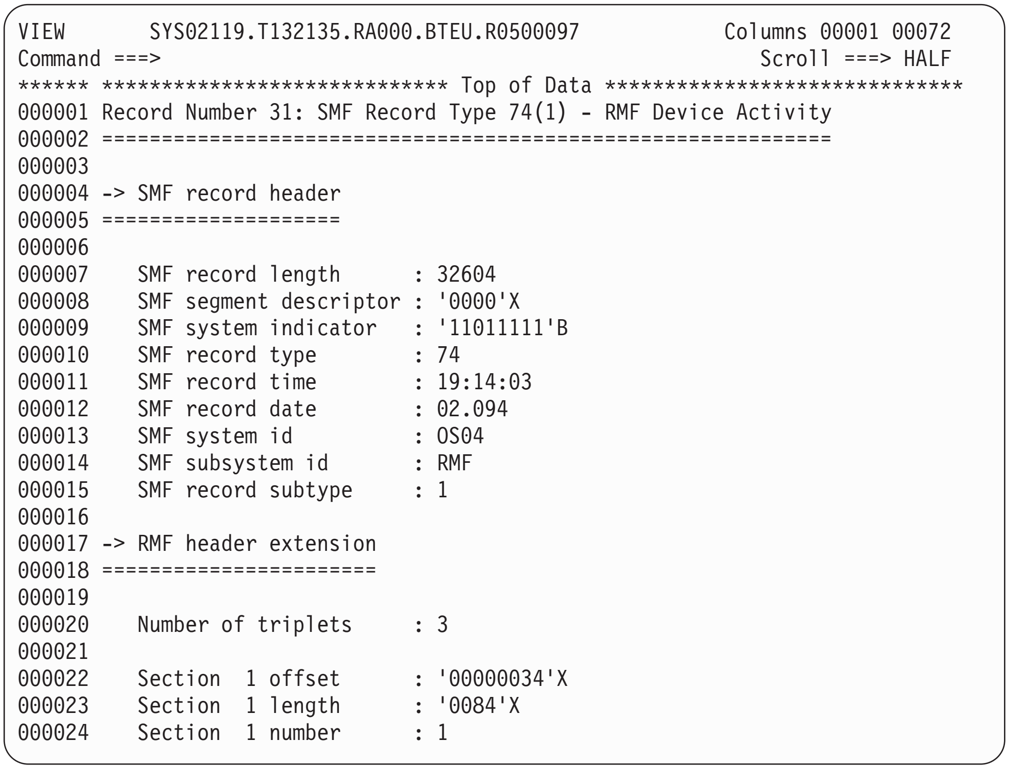
```
VIEW       SYS02119.T130441.RA000.BTEU.R0500089            Columns 00001 00072
Command ===>                                                  Scroll ===> HALF
****** ***************************** Top of Data ******************************
000001 1z/OS V1R2 RMF ERBMFSCN Version 6 (28 May 1999) - SCAN SMF dataset
000002
000003  SMF dataset characteristics:
000004  RECFM    : VBS
000005  LRECL    : 32760
000006  BLKSIZE  : 27000
000007  DATASETS : 1
000008  DSNAME(S): SYS1.SW.SMFIO
000009  DATE/TIME: 2002 April 28     13:04:41.580
000010
000011
000012 1Rec-Num Type    RecLn SMFDate  SMFTime  RMFDate  RMFTime  Int-Len   SMF
000013  ------- ------- ----- ----------------- ----------------- --------- ---
000014        1 002        18 2002.117 14:19:01                             VSP
000015        2 074.002 15064 2002.094 19:14:00 2002.094 18:59:00 15:00:000 OS0
000016        3 074.003   412 2002.094 19:14:00 2002.094 18:59:00 15:00:000 OS0
000017        4 073.001 18680 2002.094 19:14:00 2002.094 18:59:00 15:00:090 OS0
000018        5 074.004  1912 2002.094 19:14:00 2002.094 18:59:00 15:00:000 OS0
000019        6 074.004  3816 2002.094 19:14:00 2002.094 18:59:00 15:00:000 OS0
000020        7 078.003 29976 2002.094 19:14:00 2002.094 18:59:00 15:00:090 OS0
000021        8 074.001 15520 2002.094 19:14:03 2002.094 18:59:00 15:00:090 OS0
000022        9 074.001 32604 2002.094 19:14:03 2002.094 18:59:00 15:00:090 OS0
```

Figure 1-3. ERBSCAN - Display RMF Record List

The function ERBSHOW is part of ERBSCAN to format a specific record. You call ERBSHOW with specifying a record number, for example:

```
COMMAND===> ERBSHOW 31
```

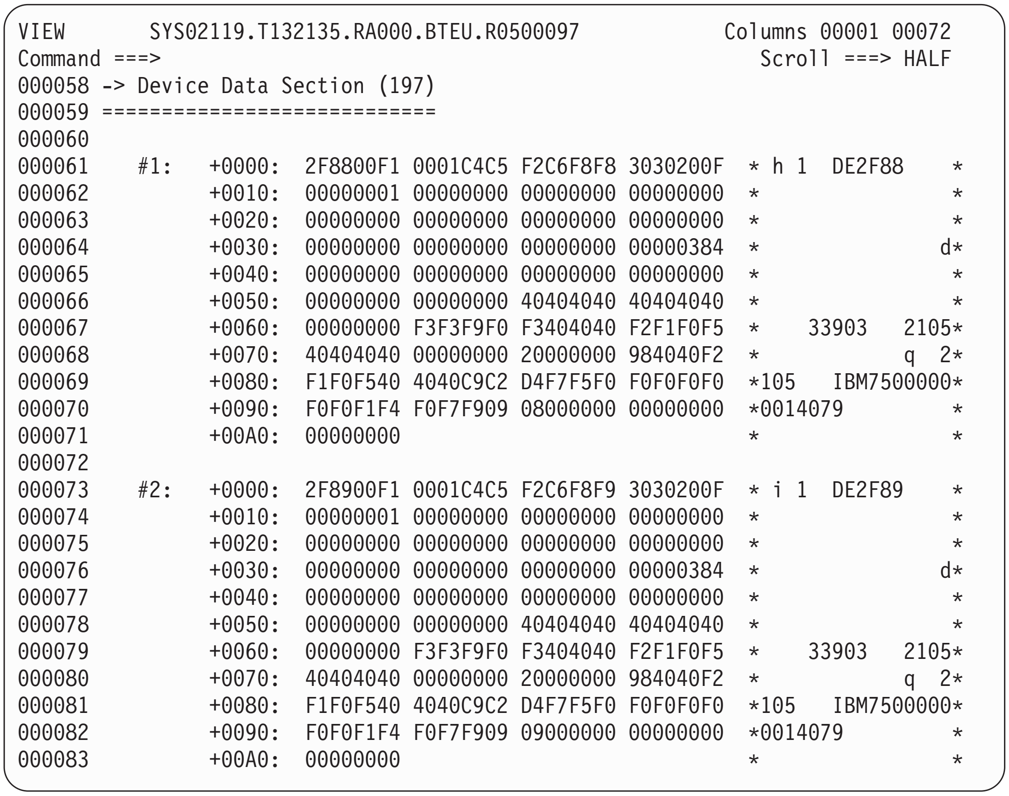
This leads to the display of the specified record showing the different sections as they are defined in the record:



```
VIEW       SYS02119.T132135.RA000.BTEU.R0500097            Columns 00001 00072
Command ===>                                                  Scroll ===> HALF
****** ***************************** Top of Data ******************************
000001 Record Number 31: SMF Record Type 74(1) - RMF Device Activity
000002 =============================================================
000003
000004 -> SMF record header
000005 ====================
000006
000007    SMF record length      : 32604
000008    SMF segment descriptor : '0000'X
000009    SMF system indicator   : '11011111'B
000010    SMF record type        : 74
000011    SMF record time        : 19:14:03
000012    SMF record date        : 02.094
000013    SMF system id          : OS04
000014    SMF subsystem id       : RMF
000015    SMF record subtype     : 1
000016
000017 -> RMF header extension
000018 =======================
000019
000020    Number of triplets     : 3
000021
000022    Section  1 offset      : '00000034'X
000023    Section  1 length      : '0084'X
000024    Section  1 number      : 1
```

Figure 1-4. ERBSHOW - Display RMF Record Header

Scrolling forward leads to the different sections of the record.



```
VIEW       SYS02119.T132135.RA000.BTEU.R0500097            Columns 00001 00072
Command ===>                                                  Scroll ===> HALF
000058 -> Device Data Section (197)
000059 ============================
000060
000061    #1:   +0000:  2F8800F1 0001C4C5 F2C6F8F8 3030200F  * h 1  DE2F88    *
000062          +0010:  00000001 00000000 00000000 00000000  *                *
000063          +0020:  00000000 00000000 00000000 00000000  *                *
000064          +0030:  00000000 00000000 00000000 00000384  *               d*
000065          +0040:  00000000 00000000 00000000 00000000  *                *
000066          +0050:  00000000 00000000 40404040 40404040  *                *
000067          +0060:  00000000 F3F3F9F0 F3404040 F2F1F0F5  *    33903   2105*
000068          +0070:  40404040 00000000 20000000 984040F2  *            q  2*
000069          +0080:  F1F0F540 4040C9C2 D4F7F5F0 F0F0F0F0  *105   IBM7500000*
000070          +0090:  F0F0F1F4 F0F7F909 08000000 00000000  *0014079         *
000071          +00A0:  00000000                             *                *
000072
000073    #2:   +0000:  2F8900F1 0001C4C5 F2C6F8F9 3030200F  * i 1  DE2F89    *
000074          +0010:  00000001 00000000 00000000 00000000  *                *
000075          +0020:  00000000 00000000 00000000 00000000  *                *
000076          +0030:  00000000 00000000 00000000 00000384  *               d*
000077          +0040:  00000000 00000000 00000000 00000000  *                *
000078          +0050:  00000000 00000000 40404040 40404040  *                *
000079          +0060:  00000000 F3F3F9F0 F3404040 F2F1F0F5  *    33903   2105*
000080          +0070:  40404040 00000000 20000000 984040F2  *            q  2*
000081          +0080:  F1F0F540 4040C9C2 D4F7F5F0 F0F0F0F0  *105   IBM7500000*
000082          +0090:  F0F0F1F4 F0F7F909 09000000 00000000  *0014079         *
000083          +00A0:  00000000                             *                *
```

Figure 1-5. ERBSHOW - Display Device Data Section

Obtaining SMF record data directly

Programming Interface information

The RMF data interface service for Monitor II allows you to directly access SMF record data from storage in real time, rather than through SMF. Application programs can easily access SMF record data. The service provides easy access to SMF data for application programs. SMF record type 79, and the Monitor II header information for system CPU utilization and system demand paging rate, are supported.

To use the RMF data interface service, invoke the module ERBSMFI with the registers and parameters described in "Parameter list contents".

Note: Do not link the module ERBSMFI into your application program. Code the program to call ERBSMFI at run time. How to do this depends on the programming language you use:

- In Assembler, use LOAD or LINK macros
- In PL/I, use FETCH and RELEASE
- In C, use the fetch built-in function

The service returns only *one* record to the caller, which contains all the data. There is no 32K size limit; that is, the record is not broken up into 32K records.

The caller must be in 31-bit addressing mode and can run unauthorized.

Note that for some of the records, Monitor I must be running. These are as follows:
Subtype 8 - Transaction activity data
Subtype 9 - Device activity
Subtype 11 - Paging activity
Subtype 14 - I/O queuing activity

For more information about SMF record type 79, see "SMF record type 79" on page 4-11.

Registers at entry

The contents of the registers on entry to this service are:

Register	Contents
0	Not used
1	Parameter list address
2-12	Not used
13	Standard save area address
14	Return address
15	Entry point address of ERBSMFI

Parameter list contents

The parameter list passed by the caller to the RMF Monitor II data interface service contains nine fullword pointers, which contain the addresses of the following parameters:

Obtaining SMF records

Parameter 1 Fullword. Request type:

1 Parameter list contains 7 parameters

2 Parameter list contains 8 parameters

3 Parameter list contains 9 parameters

Parameter 2 Fullword. SMF record type requested, of which only type 79 is supported.

Parameter 3 Fullword. SMF record subtype requested.

Parameter 4 Buffer where the SMF record output is returned. Only one record is returned. See "Output" on page 1-11.

Parameter 5 Fullword. Length of the SMF record buffer.

To determine valid record lengths, see *z/OS MVS System Management Facilities (SMF)*. For address space related SMF record type 79 subtypes 1, 2, and 5, you must provide enough space for ASVTMAXU users. RMF does not return partial data. For other SMF record type 79 subtypes, RMF returns partial data if the buffer is not long enough.

Parameter 6 Fullword. Returns the system CPU utilization.

Parameter 7 Fullword. Returns the system demand paging rate.

Parameter 8 Input area which can hold the options used to generate the Monitor II reports.

The area starts with a 2-byte length field followed by the options. If the length field is initialized with 0, the default options are taken:

Subtype	Command
1	ASD(A,A,A)
2	ARD(A,A,A)
3	SRCS
4	SPAG
5	ASRM(A,A,A)
6	SENQR(ALLVSER)
7	SENQ(S)
8	TRX(ALLPGN)
9	DEV(NUMBER(0:FFFF))
10	DDMN
11	PGSP(PAGE)
12	CHANNEL
14	IOQUEUE(DASD)
15	ILOCK

This parameter allows you to pass certain report options to the Monitor II data gatherer when parameter 1 contains the request type **2** or **3**. Please, refer to the *z/OS RMF User's Guide*, chapter "*Snapshot Reporting with Monitor II*", for other options. Use the display-session syntax described there.

Parameter 9 Fullword. Returns the MVS/SRM CPU utilization.

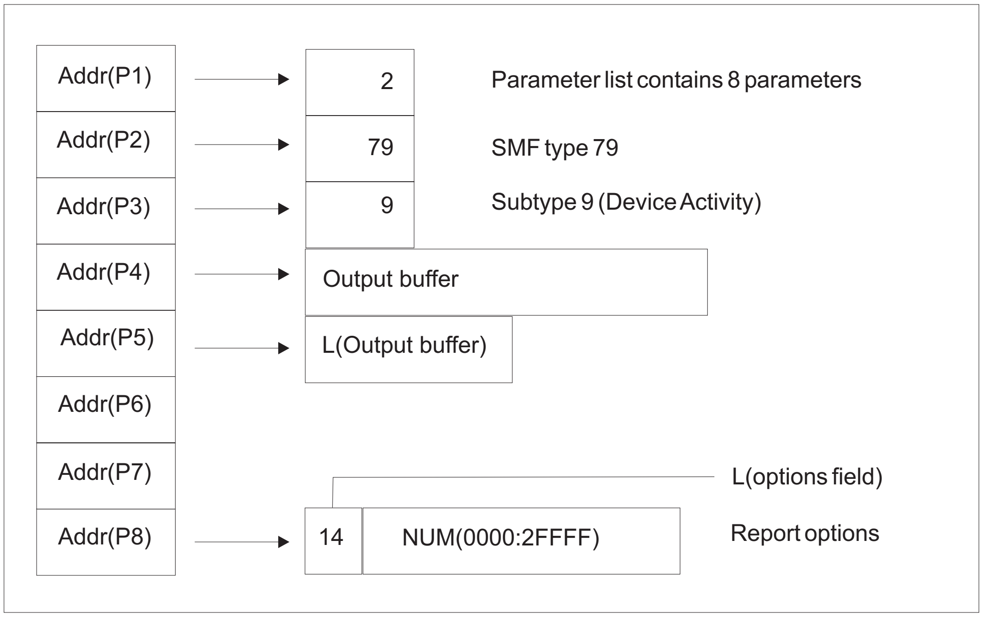
This parameter is accepted for request type **3** only.

Example:

To generate data for the Monitor II Device Activity report for all addresses in the range 0000 to 2FFF, you would have to issue the command:

```
DEV NUM(0000:2FFF)
```

You can specify this command with the following parameter list:



Output

The following are output considerations for parameters 4, 6, 7, and 9:

Parameter 4 Contains the one SMF record that is returned with all of the data for the system. The SMFxxLEN field contains the length of the input buffer, not the actual length of the record. If the buffer is over 64K, the record contains X'FFFF'. If necessary, you can calculate the actual length of the record from the descriptor fields in the record. The date and time fields (SMF79DTE and SMF79TME fields, respectively) contain zeroes.

In case RMF was not started since the last IPL, the following fields are set to these values:

SMF79IML	X’FF’
SMF79PTN	X’FF’
SMF79FLG	LSB (bit 7) off
SMF79PRF	Bits 1 and 2 off

Parameter 6 Contains the current average processor utilization percent as a binary fullword in the area provided. If RMF cannot determine the CPU utilization percent on a PR/SM system because the Monitor I CPU report is not active, RMF returns a value of –1 (FFFFFFFF).

Parameter 7 Contains the page-ins per second rate as a binary fullword in the area provided. This rate is for demand paging to DASD only. It excludes swap-ins, VIO (virtual input/output), and hiperspaces.

Parameter 9 Contains the MVS view of the CPU utilization if Monitor I CPU gathering is active. Otherwise it is filled with the SRM view of the CPU utilization (source is CCVUTILP).

Return codes

Upon return from this service, register 15 provides return codes listed in Table 1-1.

Table 1-1. Return Codes for the Monitor II Data Interface Service

Return Code (Decimal)	Description
0	Normal completion, data returned.
4	Incorrect syntax in parameter string.
8	Incorrect entry code (internal error in ERBSMFI).
16	Data is currently not available. It may be available at another time. Try again later.
20	Recovery environment could not be established.
24	Syntax error.
28	Data could not all fit in the buffer. Part of the data is returned. To get complete data, use a longer SMF buffer.
32	Data is not available; Monitor I gatherer is not active.
36	Data is reinitialized; Monitor I interval ended.
40	Data is not available. System resource manager's (SRM) store channel path status (STCPS) facility is not active.
44	Data is not available. System is in goal mode.
48	No transaction data available.
60	Invalid I/O measurement level.
100	Input record type or subtype is not valid.
104	No data is returned; SMF record buffer is too short.
108	Request type is not known.
112	ESTAE routine had control.
116	RMF is not enabled to run on this system.
120	Service IFAEDREG or IFAEDDRG for registration or deregistration returned with a code greater than 4.

Coded example

The following Assembler code example calls the RMF Monitor II data interface service to obtain SMF record type 79 subtype 2 (address space resource data).

```
         ICTL     1,71,20
         PRINT    ON,GEN
EXSMFI   CSECT
         STM      R14,R12,12(R13)     Save entry regs
         LR       R12,R15             Set base from entry point
         USING    EXSMFI,R12          Tell asmblr of prcdr base
         LA       R2,SAVEAREA         Ptr to save area
         ST       R13,4(,R2)          Save old save in new area
         ST       R2,8(,R13)          Save new as forward of last
         LR       R13,R2              Point at new
* Get storage for SMF record buffer
         LA       R3,R792RLEN         Length of data section
         L        R4,CVTPTR           Address of CVT
         USING    CVT,R4
         L         R5,CVTASVT          ASVT address
         USING     ASVT,R5
         M         R2,ASVTMAXU         Multiply by maximum users
         DROP      R4                  CVT no longer needed
         DROP      R5                  ASVT no longer needed
         A         R3,HDRLEN           Add length of record headers
         SR        R4,R4               Subpool 0
         GETMAIN   RU,LV=(3),SP=(4)    Get storage
         ST        R1,BUFFER           Buffer address to parm list
         ST        R3,BUFLEN           Length to parm list
* Call ERBSMFI to create the record
         LA        R1,PARMLIST         Parameter to reg 1
         LINK      EP=ERBSMFI
*
* Check the return code and process the record here
*
         L         R2,BUFFER           Get ptr to buffer start
         L         R3,BUFLEN           Get buffer length
         SR        R4,R4               Subpool zero
         FREEMAIN  RU,LV=(3),A=(2),SP=(4)
         L         R13,4(,R13)         Point at old save area
         SR        R15,R15             Set return code
         L         R14,12(,R13)        Restore return register
         LM        R0,R12,20(R13)      Restore all the rest
         BR        R14                 Return to caller
SAVEAREA DS        CL72                Save area
PARMLIST DC       A(REQTYPE)           Pointer to request type
         DC       A(RECTYPE)           Pointer to record type
         DC       A(SUBTYPE)           Pointer to subtype
BUFFER   DS        A                   Pointer to output buffer
         DC       A(BUFLEN)            Pointer to buffer length
         DC       A(CPUUTL)            Pointer to CPU utilization
         DC       A(DPR)               Pointer to demand paging rate
REQTYPE  DC       F'1'                 Request type
RECTYPE  DC       F'79'                Record type 79
SUBTYPE  DC       F'2'                 Subtype for ARD report record
BUFLEN   DS        F                   Length of SMF record buffer
CPUUTL   DS        F                   Return area for CPU util.
DPR      DS        F                   Return area for demand paging
HDRLEN   DC        A(HLEN+PLEN+CLEN)   Header length
*                                                                     *
***********************************************************************
*        Patch Area                                                   *
***********************************************************************
PATCH    DC        64S(*)
*
         LTORG
*
         PRINT     NOGEN
* SMF record 79 mapping
         ERBSMFR   79
* Record lengths
SMF79HDR DSECT
HLEN     EQU       *-SMF79HDR
SMF79PRO DSECT
PLEN     EQU       *-SMF79PRO
R79CHL   DSECT
CLEN     EQU       *-R79CHL
EXSMFI   CSECT
* System control block mappings
         CVT       DSECT=YES,LIST=NO
         IHAASVT   DSECT=YES,LIST=NO
* Registers
R0       EQU       0
R1       EQU       1
R2       EQU       2
R3       EQU       3
```

```
R4       EQU      4
R5       EQU      5
R6       EQU      6
R7       EQU      7
R8       EQU      8
R9       EQU      9
R10      EQU      10
R11      EQU      11
R12      EQU      12
R13      EQU      13
R14      EQU      14
R15      EQU      15
         END      EXSMFI
```

End of Programming Interface information

Chapter 2. RMF sysplex data services

Data Access Across the Sysplex

The information in this chapter describes callable services that RMF provides to enable you to access sysplex data:

- ERBDSQRY - RMF Query Available Sysplex SMF Data Service
- ERBDSREC - RMF Request Sysplex SMF Record Data Service
- ERB2XDGS - RMF Monitor II Sysplex Data Gathering Service
- ERB3XDRS - RMF Monitor III Sysplex Data Retrieval Service

This chapter describes the CALL statements that invoke RMF sysplex data services. Each description includes a syntax diagram, parameter descriptions, and return code and reason code explanations with recommended actions. Return codes and reason codes are shown in decimal.

How to call sysplex data services

To use RMF sysplex data services, you issue CALLs that invoke the appropriate data service program. Each service program performs one or more functions and requires a set of parameters coded in a specific order on the CALL statement.

Do not link the data-services modules into your application program. Code the program to call the modules at run time. How you do this depends on the programming language you use:

- In Assembler, use LOAD or LINK macros
- In PL/I, use FETCH and RELEASE
- In C, use the fetch built-in function

The RMF supplied samplib contains three sample programs (written in C) that invoke these services:

- ERBDSMP1 calls ERBDSQRY/ERBDSREC
- ERBDSMP2 calls ERB2XDGS
- ERBDSMP3 calls ERB3XDRS

You might take one of the above sample programs as base for your own program MYERBSRV which you will code according to your requirements.

Example: The sample program ERBDSMP2 specifies ERBDSMX2 as exit. This program (an assembler program) is also part of the samplib and might be used as example on how to write an exit program.

Typically, one would not use sample exit ERBDSMX2 when running MYERBSRV, but would either use the RMF supplied exit ERB2XSMF which returns the complete SMF type 79 records, or one might write an own exit routine for a data reduction.

For ERB3XDRS (ERBDSMP3 sample program), the RMF supplied exit routine provided is ERB3XSOS (returns the complete Monitor III Set-of-Samples).

A sample JCL for compile, link and go could be set up as follows:

```
//    job record
//*
//PROCLIB  JCLLIB  ORDER=CBC.SCBCPRC
//*
//        EXEC EDCCLG,
//        INFILE='SYS1.SAMPLIB(ERBDSMP2)',
//        OUTFILE='loadlib(ERBDSMP2),DISP=SHR',
//        CPARM='OPTFILE(DD:OPTS)'
//COMPILE.OPTS DD *
SEARCH('SYS1.SAMPLIB')
SOURCE,NOLIST,OPTIMIZE,NOXREF,GONUMBER
```

ERBDSQRY - RMF Query available sysplex SMF data service

Programming Interface information

Call ERBDSQRY to request a directory of SMF record data available in the RMF Data Buffers on each system in the sysplex.

Write the CALL for ERBDSQRY as shown, coding all parameters in the specified order. Ensure that the values you assign to the parameters are in the format shown.

Table 2-1. ERBDSQRY Service

CALL ERBDSQRY	(answer_area_addr ,answer_area_alet ,answer_area_length ,request_type ,start_time ,end_time ,smf_record_type_info ,smf_record_type_list ,smf_system_name_info ,smf_system_name_list ,time_out ,return_code ,reason_code)

answer_area_addr

Specifies the address of the area where RMF returns the requested information. The area can be in the caller's primary address space or in an address or data space addressable through a public entry on the caller's Dispatchable Unit Access List (DU-AL).

Define *answer_area_addr* as pointer variable of length 4.

,answer_area_alet

Specifies the ALET of the answer area provided on the *answer_area_addr* parameter. If the area resides in the caller's primary address space, *answer_area_alet* must be 0.

Define *answer_area_alet* as unsigned integer variable of length 4.

,answer_area_length

Specifies the length of the answer area provided on the *answer_area_addr* parameter. If you do not provide enough length, RMF sets a return code and reason code, and places the length you need in the *answer_area_length* parameter.

Define *answer_area_length* as an unsigned integer variable of length 4.

,request_type

Specifies the ERBDSQRY request type. Specify one of the following values:

SMF Request information about SMF records of any type and subtype. Information will be returned about all SMF records whose time information, specified in the SMF record header, is within the time interval specified in the *start_time* and *end_time* parameters, that is:

```
C2S(start_time) ≤ (SMFxxDTE;SMFxxTME) ≤ C2S(end_time)
```

where C2S is the conversion function from character to SMF date and time format.

Note: This is the time the record was presented to SMF. For RMF-gathered data, it does not necessarily coincide exactly with the interval end time of the data collection interval.

The directory entries returned by ERBDSQRY contain the SMF record header plus a record token.

RMF Request information about SMF records of any RMF type and subtype. Information will be returned about all SMF records whose projected RMF measurement interval end time, specified in the RMF product section, is within the time interval specified in the *start_time* and *end_time* parameters, that is:

```
C2T(start_time) ≤ (SMFxxGIE + SMFxxLGO) ≤ C2T(end_time)
```

where C2T is the conversion function from character to time-of-day (store clock) format.

Note: This is a theoretical value, it may not coincide with the actual RMF measurement interval (also part of the RMF product section of the SMF record).

The directory entries returned by ERBDSQRY contain SMF record header, RMF measurement interval information, plus a record token.

See "ERBDSQRY data section layout" on page 2-23.

Define *request_type* as character variable of length 3.

,start_time

Specifies the beginning of the time interval for which information is requested.

Define *start_time* as character variable of length 14 in the ″sorted″ format:

yyyy	mm	dd	hh	mm	ss

If you want to omit this information, pass a value of 14 blanks. It will then default to the ″oldest″ SMF time found in any of the RMF Data Buffers at the time the service is called.

,end_time

Specifies the date and time of the end of the time interval information is requested for.

Define *end_time* as character variable of length 14 in the same ″sorted″ format as *start_time*.

If you want to omit this information, pass a value of 14 blanks. It will then default to the ″newest″ SMF time found in any of the RMF Data Buffers at the time the service is called.

,smf_record_type_info

Specifies the type of the list of SMF record types provided on the *smf_record_type_list* parameter. Specify one of the following values:

INCLUDE
The list of SMF record types provided on the *smf_record_type_list* parameter is an inclusion list. Information is requested for the listed SMF record types.

EXCLUDE
The list of SMF record types provided on the *smf_record_type_list* parameter is an exclusion list. Information is requested for all but the listed SMF record types.

ALL Information is requested for all SMF record types. The list of SMF record types provided on the *smf_record_type_list* parameter must start with an unsigned integer variable of length 4 set to a value of 0 (zero).

Define *smf_record_type_info* as a character variable of length 7. If you specify ALL, pad the string on the right with 4 blanks.

,smf_record_type_list

Specifies the list of SMF record types for which information is requested.

Define *smf_record_type_list* as an unsigned integer variable of length 4 (#rtypes) followed by an array of pairs of unsigned integers of length 2 (rt1... and st1...). The variable *#rtypes* specifies the number of array elements. Give *#rtypes* the value 0 (zero) to obtain information for all record types. The first number of each pair (rt1...) specifies the record type, and the second number of each pair (st1...) specifies the record subtype. For record types without subtypes, specify a subtype of 0.

Note: If you have specified **RMF** for *request_type*, record types outside the range 70 to 79 are ignored.

#rtypes	rt1	st1	rt2	st2	…	…

,smf_system_name_info

Specifies the type of the list of SMF system names provided on the *smf_system_name_list* parameter. Specify one of the following values:

INCLUDE
The list of SMF system names provided on the *smf_system_name_list* parameter is an inclusion list. Information is requested for systems with the listed SMF system names.

EXCLUDE
The list of SMF system names provided on the *smf_system_name_list* parameter is an exclusion list. Information is requested for all systems in the sysplex excluding the systems with the listed SMF system names.

ALL Information is requested for all systems in the sysplex. The list of SMF system names provided on the *smf_system_name_list* parameter must start with an unsigned integer variable of length 4 set to a value of 0 (zero).

The list of SMF system names provided on the *smf_system_name_list* parameter is ignored. Information is requested for all systems in the sysplex.

Define *smf_system_name_info* as a character variable of length 7. If you specify ALL, pad the string on the right with 4 blanks.

,smf_system_name_list

Specifies the list of SMF system names information is requested for.

Define *smf_system_name_list* as an unsigned integer variable of length 4 that specifies the number of array elements, followed by an array of character variables of length 4.

#snames	#sn1	#sn2	…

,time_out

Specifies a time interval in seconds. If the time interval expires during the processing of the service, RMF returns to the caller with a corresponding return code and reason code and partial data.

Define *time_out* as a positive unsigned integer of length 4. Any other value will be overridden by a default value of 60.

,return_code

When ERBDSQRY completes, *return_code* contains the return code.

Define *return_code* as an unsigned integer variable of length 4.

For details, see “Return codes and reason codes” on page 2-17.

,reason_code

When ERBDSQRY completes, *reason_code* contains the reason code.

Define *reason_code* as an unsigned integer variable of length 4.

For details, see “Return codes and reason codes” on page 2-17.

End of Programming Interface information

ERBDSREC - RMF Request sysplex SMF record data service

Programming Interface information

Call ERBDSREC to request SMF record data from the RMF Data Buffers on each system in the sysplex. For each requested SMF record, include the record token, obtained from an earlier call of ERBDSQRY, on the list of record tokens passed as parameter to ERBDSREC.

Write the CALL for ERBDSREC as shown, coding all parameters in the specified order. Ensure that the values you assign to the parameters are in the format shown.

Table 2-2. ERBDSREC Service

CALL ERBDSREC	(answer_area_addr ,answer_area_alet ,answer_area_length ,rmf_record_token_list ,time_out ,return_code ,reason_code)

answer_area_addr

Specifies the address of the area to which RMF returns the requested information. The area can be in the caller's primary address space or in an address or data space addressable through a public entry on the caller's Dispatchable Unit Access List (DU-AL).

Define *answer_area_addr* as a pointer variable of length 4.

,answer_area_alet

Specifies the ALET of the answer area provided on the *answer_area_addr* parameter. If the area resides in the caller's primary address space, *answer_area_alet* must be 0.

Define *answer_area_alet* as an unsigned integer variable of length 4.

,answer_area_length

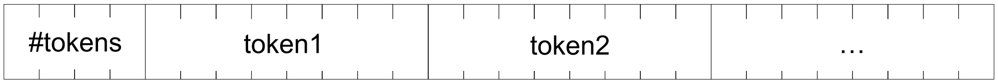
Specifies the length of the answer area provided on the *answer_area_addr* parameter. If you do not provide enough length, RMF sets a return code and reason code, and places the length you need in the *answer_area_length* parameter.

Define *answer_area_length* as unsigned integer variable of length 4.

,rmf_record_token_list

Specifies the list of record tokens for the requested SMF records.

Define *rmf_record_token_list* as an unsigned integer variable of length 4 that specifies the number of array elements, followed by an array of character of length 8.



,time_out

Specifies a time interval in seconds. If the time interval expires during the processing of the service, RMF returns to the caller with a corresponding return code and reason code and partial data.

Define *time_out* as a positive unsigned integer of length 4. Any other value will be overridden by a default value of 60.

,return_code

When ERBDSREC completes, *return_code* contains the return code.

Define *return_code* as an unsigned integer variable of length 4.

For details, see "Return codes and reason codes" on page 2-17.

,reason_code

When ERBDSREC completes, *reason_code* contains the reason code.

Define *reason_code* as an unsigned integer variable of length 4.

For details, see "Return codes and reason codes" on page 2-17.

End of Programming Interface information

ERB2XDGS - RMF Monitor II sysplex data gathering service

Programming Interface information

Call ERB2XDGS to request Monitor II data according to the specified SMF record type 79 (Monitor II) subtype.

Write the CALL for ERB2XDGS as shown, coding all parameters in the specified order. For parameters that ERB2XDGS uses to obtain input values, assign values that are acceptable to ERB2XDGS.

Table 2-3. ERB2XDGS Service

CALL ERB2XDGS	(answer_area_addr ,answer_area_alet ,answer_area_length ,system_name ,data_gatherer_parm ,data_gatherer_parm_length ,exit_name ,exit_parm ,exit_parm_length ,time_out ,return_code ,reason_code)

answer_area_addr

Specifies the address of the area where RMF returns the requested information. The area can be in the calling program's primary address space, or in an address or data space addressable through a public entry on the calling program's dispatchable unit access list (DU-AL).

Define *answer_area_addr* as pointer variable of length 4.

,answer_area_alet

Specifies the ALET of the answer area provided on the *answer_area_addr* parameter. If the area resides in the calling program's primary address space, *answer_area_alet* must be 0.

Define *answer_area_alet* as unsigned integer variable of length 4.

,answer_area_length

Specifies the length of the answer area provided on the *answer_area_addr* parameter. If you do not provide enough space, RMF lets you know how much space you should have provided. The *answer_area_length* input/output parameter contains the length needed for the complete data.

Define *answer_area_length* as unsigned integer variable of length 4.

,system_name

Specifies the name of the system for which you are requesting information. This is the four character SMF system identification (SID). ***ALL** specifies that the request is to be sent to **all** systems in the sysplex. However, only the systems with a running Monitor II data gatherer session are able to return the requested data.

Define *system_name* as character variable of length 4.

,data_gatherer_parm

Specifies the parameters for the Monitor II data gatherer on each system.

Define *data_gatherer_parm* as structure variable of variable length. The layout of the parameter area is as follows:

rty	sty	dg_options ...

where:

rty Specifies the SMF record type of the requested Monitor II data.

Define *rty* as unsigned integer variable of length 2.

sty Specifies the SMF record subtype of the requested Monitor II data.

Define *sty* as unsigned integer variable of length 2.

dg_options

Specifies options for the Monitor II data gatherer for the specified SMF record type and subtype.

Define *dg_options* as character variable of variable length, maximum 32.

You find a list of all subtypes in “Overview” on page 1-2.

Example

You want to receive data that is equivalent to the Monitor II command

```
SENQ D
```

This requires the following values for this parameter:

rty	SMF record type - **79**
sty	SMF record subtype for the `SENQ` - **07**
dg_options	Command option - **D**

This results in the value ’7907D’ for the data gatherer parameter.

,data_gatherer_parm_length

Specifies the length of the parameter string *data_gatherer_parm*.

Define *data_gatherer_parm_length* as unsigned integer variable of length 4.

,exit_name

Specifies the name of a data reduction exit routine that is invoked by RMF on each system from which data is requested. After the Monitor II data has been retrieved by RMF, this exit may call selected areas from the data to the answer area provided by RMF. These data areas are then combined into the answer area provided by the caller on the requesting system.

The data reduction exit routine ERB2XSMF, provided by IBM, copies the complete data gathered by the Monitor II data gatherer (SMF record type 79) to the answer area. ERB2XSMF has no exit parameters.

Define *exit_name* as character variable of length 8.

,exit_parm

Specifies a parameter string that may be passed to the routine specified in *exit_name*. Use this parameter to control the selection of Monitor II data areas to be returned to the caller.

Define *exit_parm* as character variable of variable length, maximum 32768.

,exit_parm_length

Specifies the length of the parameter string *exit_parm* that is passed to the routine specified in *exit_name*.

Define *exit_parm_length* as unsigned integer variable of length 4.

,time_out

Specifies a time interval in seconds. If this time interval expires during the processing of the service, RMF returns to the caller with a corresponding return and reason code and partial data.

Define *time_out* as unsigned integer variable of length 4.

The specification of a non-positive value will cause RMF to use a default value of 60.

,return_code

When ERB2XDGS completes, *return_code* contains the return code.

Define *return_code* as unsigned integer variable of length 4.

Return Codes and Reason Codes are explained under “Return codes and reason codes” on page 2-17.

,reason_code

When ERB2XDGS completes, *reason_code* contains the reason code.

Define *reason_code* as unsigned integer variable of length 4.

Return Codes and Reason Codes are explained in “Return codes and reason codes” on page 2-17.

End of Programming Interface information

ERB2XDGS data reduction exit routines

Programming Interface information

The exit routine specified in the **exit_name** parameter of the ERB2XDGS service is invoked on each system to which the ERB2XDGS request was directed. The routine is assumed to have the following attributes:

Location:	JPA
State:	Problem
Key:	Any
Amode:	31
Rmode:	Any
Dispatchable unit mode:	Task
Address space control mode:	AR
Cross Memory Mode:	PASN=SASN=HASN

Serialization:	Enabled, unlocked
Type:	Reentrant, Refreshable

The exit is called by RMF as shown, with the parameters in the specified order.

Table 2-4. ERB2XDGS Exit Routine

CALL exit_name	(answer_area_addr ,answer_area_alet ,answer_area_length ,output_area_length ,input_data_address ,exit_parm ,exit_parm_length)

answer_area_addr

Specifies the address of the area where the exit routine may return the selected information. The area resides in a data space owned by the RMF address space.

Answer_area_addr is defined as pointer variable of length 4.

,answer_area_alet

Specifies the ALET of the answer area provided on the *answer_area_addr* parameter.

Answer_area_alet is defined as unsigned integer variable of length 4.

,answer_area_length

Specifies the length of the answer area provided on the *answer_area_addr* parameter. RMF provides an answer area in the length of the answer area the caller provided to ERB2XDGS, rounded to the next multiple of 4096. However, the data returned by the data reduction exit routine must fit into the answer area the caller provided to ERB2XDGS, including the common header and data headers created by RMF.

Answer_area_length is defined as unsigned integer variable of length 4.

,output_area_length

Specifies the length of the data that the exit routine provided. If this value is larger than *answer_area_length*, a return and reason code are set, indicating that the length of the answer area was not sufficient.

Output_area_length is defined as unsigned integer variable of length 4 and **must be set by the exit routine**.

,input_area_address

Specifies the address of the SMF record type 79 image in storage.

Input_area_address is defined as pointer variable of length 4.

,exit_parm

Specifies the parameter that has been provided for the exit routine by the caller of ERB2XDGS.

Exit_parm is defined as character variable of variable length.

,exit_parm_length

Specifies the length of the parameter string *exit_parm* that was passed to the exit routine.

Exit_parm_length is defined as unsigned integer variable of length 4.

End of Programming Interface information

ERB3XDRS - RMF Monitor III sysplex data retrieval service

Programming Interface information

Call ERB3XDRS to request a set-of-samples of Monitor III data from to the specified date and time range.

Write the CALL for ERB3XDRS as shown, coding all parameters in the specified order. For parameters that ERB3XDRS uses to obtain input values, assign values that are acceptable to ERB3XDRS.

Table 2-5. ERB3XDRS Service

CALL ERB3XDRS	(answer_area_addr ,answer_area_alet ,answer_area_length ,system_name ,data_retrieval_parm ,data_retrieval_parm_length ,exit_name ,exit_parm ,exit_parm_length ,time_out ,return_code ,reason_code)

answer_area_addr
Specifies the address of the area to which RMF returns the requested information. The area can be in the calling program's primary address space or in an address or data space addressable through a public entry on the calling program's dispatchable unit access list (DU-AL).

Define *answer_area_addr* as pointer variable of length 4.

,answer_area_alet
Specifies the ALET of the answer area provided on the *answer_area_addr* parameter. If the area resides in the calling program's primary address space, *answer_area_alet* must be 0.

Define *answer_area_alet* as unsigned integer variable of length 4.

,answer_area_length
Specifies the length of the answer area provided on the *answer_area_addr* parameter. If you do not provide enough space, RMF lets you know how much space you should have provided. The *answer_area_length* input/output parameter contains the length needed for the complete data.

Define *answer_area_length* as unsigned integer variable of length 4.

,system_name
Specifies the name of the system for which information is being requested. This is the four-character SMF system ID (SID). ***ALL** specifies that the request is to be sent to **all** systems in the sysplex. However, only the systems with a running Monitor III data gatherer session are able to return the requested data.

Define *system_name* as character variable of length 4.

,data_retrieval_parm

Specifies the parameters for the retrieval of Monitor III data on each system.

Define *data_retrieval_parm* as structure variable with a length of 34 bytes. This structure contains the start and end of the range for which data is requested, and parameters that define the format of the returned data. The layout of the 34-byte parameter area is as follows:

start_time	end_time	df_ssos	df_ comp

start_time

Specifies the date and time of the beginning of the time range for which information is requested.

Define *start_time* as a character variable of length 14 in “sorted” format.

yyyy	mm	dd	hh	mm	ss

If you want to omit this information, pass a value of 14 blanks. ERB3XDRS will then return information for one Monitor III MINTIME, ending with or containing the date and time specified in *end_time*. If this parameter is omitted as well, information for the latest available MINTIME is returned.

end_time

Specifies the date and time of the end of the time range for which information is requested.

Define *end_time* as character variable of length 14 in the same ″sorted″ format as *start_time*.

If you want to omit this information, pass a value of 14 blanks. ERB3XDRS will then return information for one Monitor III MINTIME, starting with or containing the date and time specified in *start_time*. If this parameter is omitted as well, information for the latest available MINTIME is returned.

df_ssos

Data format Single Set-Of-Samples - specifies whether or not the set-of-samples data should be returned as a combined set-of-samples (as opposed to a sequence of individual sets-of-samples).

YES the data is returned in a combined form, that is, the individual sets-of-samples are combined into one common set-of-samples.

NO the data is returned in individual sets-of-samples.

Define *df_ssos* as character variable of length 3. If you specify NO, pad the string on the right with a blank.

df_comp

Data format Compressed Set-Of-Samples - specifies whether or not the set-of-samples data should be returned in compressed format

YES the data is returned compressed (as it resides in the Monitor III data sets). This means that it will have to be decompressed using the RMF service ERB3RDEC.

NO the data is returned uncompressed

Define *df_comp* as character variable of length 3. If you specify NO, pad the string on the right with a blank.

,data_retrieval_parm_length

Specifies the length of the parameter string *data_retrieval_parm*.

Define *data_retrieval_parm_length* as unsigned integer variable of length 4.

,exit_name

Specifies the name of a data reduction exit routine that is invoked by RMF on each system from which data is requested. After the set-of-samples data has been retrieved by RMF, this exit may call selected areas from the set-of-samples to the answer area provided by RMF. These data areas are then combined into the answer area provided by the caller on the requesting system.

The data reduction exit routine ERB3XSOS, provided by IBM, copies the complete data retrieved from the Monitor III data gatherer (the set-of-samples data) to the answer area. ERB3XSOS has no exit parameters.

Define *exit_name* as a character variable of length 8.

,exit_parm

Specifies a parameter string that may be passed to the routine specified in *exit_name*. Use this parameter to control the selection of set-of-samples data areas that are to be returned to the caller.

Define *exit_parm* as a character variable of variable length, with a maximum of 32768.

,exit_parm_length

Specifies the length of the parameter string *exit_parm* that is passed to the routine specified in *exit_name*.

Define *exit_parm_length* as an unsigned integer variable of length 4.

,time_out

Specifies a time interval in seconds. If this time interval expires during the processing of the service, RMF returns to the caller with a corresponding return and reason code and partial data.

Define *time_out* as an unsigned integer variable of length 4.

The specification of a non-positive value will cause RMF to use a default value of 60.

,return_code

When ERB3XDRS completes, *return_code* contains the return code.

Define *return_code* as an unsigned integer variable of length 4.

Return Codes and Reason Codes are explained under “Return codes and reason codes” on page 2-17.

,reason_code

When ERB3XDRS completes, *reason_code* contains the reason code.

Define *reason_code* as an unsigned integer variable of length 4.

Return Codes and Reason Codes are explained in "Return codes and reason codes" on page 2-17.

End of Programming Interface information

ERB3XDRS data reduction exit routines

Programming Interface information

The exit routine specified in the **exit_name** parameter of the ERB3XDRS service is invoked on each system the ERB3XDRS request was directed to. The routine is assumed to have the following attributes:

Location:	JPA
State:	Problem
Key:	Any
Amode:	31
Rmode:	Any
Dispatchable unit mode:	Task
Address space control mode:	AR
Cross Memory Mode:	PASN=SASN=HASN
Serialization:	Enabled, unlocked
Type:	Reentrant, Refreshable

The exit is called by RMF as shown, with the parameters in the specified order.

Table 2-6. ERB3XDRS Exit Routine

CALL exit_name	(answer_area_addr ,answer_area_alet ,answer_area_length ,output_area_length ,input_data_address ,exit_parm ,exit_parm_length)

answer_area_addr

Specifies the address of the area to which the exit routine may return the selected information. The area resides in a data space owned by the RMF address space.

Answer_area_addr is defined as a pointer variable of length 4.

,answer_area_alet

Specifies the ALET of the answer area provided on the *answer_area_addr* parameter.

Answer_area_alet is defined as an unsigned integer variable of length 4.

,answer_area_length

Specifies the length of the answer area provided on the *answer_area_addr* parameter. RMF provides an answer area the same length as the answer area that the caller provided for ERB3XDRS, rounded to the next multiple of 4096. However, the data returned by the data reduction exit routine must fit into the answer area the caller provided for ERB3XDRS, including the common header and data headers created by RMF.

Answer_area_length is defined as an unsigned integer variable of length 4.

,output_area_length

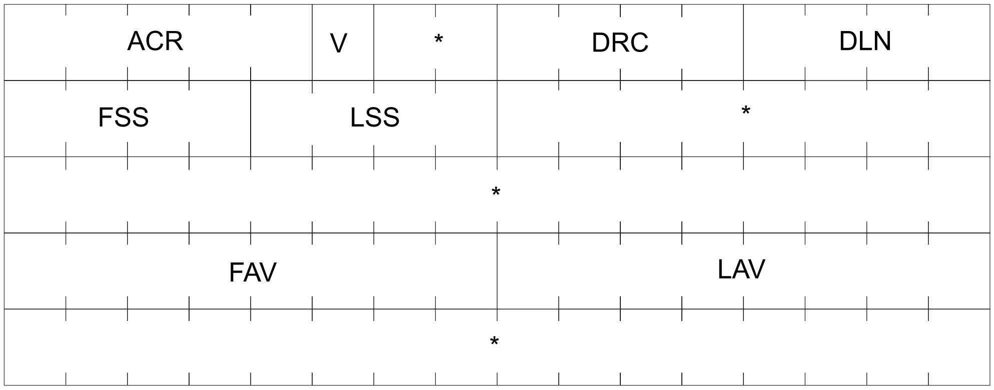
Specifies the length of the data that is provided by the exit routine. If this value is larger than *answer_area_length*, a return and reason code is set, indicating that the length of the answer area is not sufficient.

Output_area_length is defined as an unsigned integer variable of length 4 and **must be set by the exit routine**.

,input_area_address

Specifies the address of the data reduction exit input data area. This data area contains the Monitor III control block XMHG3 at offset 0, followed by zero or more sets-of-samples, each of them starting with the Monitor III control block SSHG3.

Input_area_address is defined as a pointer variable of length 4. Control block XMHG3 has the following format:



ACR (offset +00, length 5) Acronym of XMHG3, EBCDIC ″XMHG3″

V (offset +05, length 1) Version of XMHG3

DRC (offset +08, length 4) Data return code. The possible codes are:

0	Successful data retrieval
4	Time out of range
8	Area too small
12	No data available
16	Severe error

DLN (offset +12, length 4) Total data length including XMHG3 itself

FSS (offset +16, length 4) Offset from XMHG3 to first set-of-samples header SSHG3

LSS (offset +20, length 4) Offset from XMHG3 to last set-of-samples header SSHG3

FAV (offset +40, length 8) Time in STCK format of first available data

LAV (offset +48, length 8) Time in STCK format of last available data

,exit_parm
Specifies the parameter for the exit routine that has been provided by the caller of ERB3XDRS.

Exit_parm is defined as character variable of variable length.

,exit_parm_length
Specifies the length of the parameter string *exit_parm* that is passed to the exit routine.

Exit_parm_length is defined as unsigned integer variable of length 4.

End of Programming Interface information

Return codes and reason codes

Programming Interface information

When the RMF Sysplex Data Services ERBDSQRY, ERBDSREC, ERB2XDGS, and ERB3XDRS return control to your program, *return_code* contains the return code and *reason_code* contains the reason code.

Not every combination of return and reason codes applies to each of the services. The possible combinations are shown in Table 2-7 on page 2-18.

The return and reason codes are grouped into classes indicating the severity of the situation that has been recognized. The classes are:

Successful (RC=0)
The operation was successful. The requested data has been stored in the answer area provided by the calling program

Information (RC=4)
The requested data may be inconsistent (ERB3XDRS only)

Warning (RC=8)
The requested data could not be retrieved completely

Error (RC=12)
No data was returned, for example, because no RMF address space was active

Severe Error (RC=16)
The calling program invoked the service with invalid parameters or in an invalid mode

Unrecoverable Error (RC=20)
A problem has been detected within RMF processing. This code is normally accompanied by console messages, or a dump, or both. Refer to the explanations of the issued messages.

The following table identifies return code and reason code combinations, and recommends the action that you should take. Codes are decimal numbers. Applicable service routines are:

Q	ERBDSQRY
R	ERBDSREC

2 ERB2XDGS
3 ERB3XDRS

Table 2-7. RMF Sysplex Data Services Return and Reason Codes (SMF Services)

Return Code	Reason Code	Service Routine	Meaning
			Action
0	0	Q,R,2,3	**Meaning:** The operation was successful. The answer area contains the requested data.
			Action: Continue normal program execution.
8	8	-,-,-,3	**Meaning:** Warning - data could not be retrieved. For the specified date and time range, either partial data or no data at all could be retrieved by the ERB3XDRS service because time gaps have been detected in the gathered data.
			Action: Check the time range (*start_time* or *end_time*) parameters on the ERB3XDRS service and rerun the program.
8	9	-,-,-,3	**Meaning:** Warning - VSAM retrieval errors occurred. For the specified date and time range, either partial data or no data at all could be retrieved.
			Action: Check the time range (*start_time* or *end_time*) parameters on the ERB3XDRS service and rerun the program.
8	13	-,-,-,3	**Meaning:** Warning - inconsistent data returned by ERB3XDRS. The WLM service policy has changed, or the IPS values have been modified.
8	14	-,-,-,3	**Meaning:** Warning - inconsistent data returned by ERB3XDRS. The RMF cycle time has changed.
8	15	-,-,-,3	**Meaning:** Warning - inconsistent data returned by ERB3XDRS. IPL detected.
8	30	Q,R,-,-	**Meaning:** Warning - timeouts detected. Due to timeout situations, ERBDSQRY or ERBDSREC could not return all the requested information.
			Action: Request a smaller a mount of information on one call of the RMF service.
8	31	-,R,-,-	**Meaning:** Warning - no such record. One or more requested SMF records were not available for ERBDSREC, either the SMF record data was overwritten by the wrap-around management of the data buffer or it never existed.
			Action: Ensure that the elapsed time between calls to ERBDSQRY and ERBDSREC is not too large, and that a valid token list is passed to ERBDSREC.
8	35	-,-,2,-	**Meaning:** Warning - defaults taken. Due to incorrectly specified Monitor II data gatherer options on the *dg_options* parameter of the ERB2XDGS service, the data gatherer decided to use the default options.
			Action: Correct Monitor II data gatherer options and rerun the program.
8	70	Q,R,-,-	**Meaning:** Warning - answer area too small. The answer area provided by the calling program was too small for the service to return all the requested information. The variable *answer_area_length* contains the length of the answer area you should have provided for this ERBDSQRY or ERBDSREC request.
			Action: Provide an answer area large enough to contain all the requested information.
12	0	Q,R,2,3	**Meaning:** Error - RMF Sysplex Data Server is not active.
			Action: Start the local RMF address space.
12	1	Q,R,2,3	**Meaning:** Error - System(s) inactive. None of the system(s) specified for the ERBDSQRY, ERB2XDGS, or ERB3XDRS services were active in the sysplex. For ERBDSREC, none of the record tokens specified belong to SMF records collected on systems that are currently active in the sysplex.
			Action: Check the system name list (*smf_system_name_list*, for ERBDSQRY), record token list (*rmf_record_token_list*, for ERBDSREC), or the system name (*system_name*, for ERB2XDGS and ERB3XDRS) parameter and rerun the program.
12	5	-,-,2,-	**Meaning:** Error - Monitor I interval ended. The Monitor I interval ended during the Monitor II data gathering phase while processing the ERB2XDGS request.
			Action: Rerun the program.

Table 2-7. RMF Sysplex Data Services Return and Reason Codes (SMF Services) (continued)

Return Code	Reason Code	Service Routine	Meaning
			Action
12	6	-,-,2,-	**Meaning:** Error - No RMF data available. No data is currently available that matches the specification in the *data_gathering_parm* parameter of the ERB2XDGS service.
			Action: Check the parameters of ERB2XDGS and rerun the program.
12	7	-,-,2,-	**Meaning:** Error - No Monitor I data gatherer. The Monitor I data gatherer was not active. However, for the data gathering of certain SMF record subtypes (record type 79, subtypes 8, 9, 11, 13, and 14) specified for the ERB2XDGS service, an active Monitor I session is required.
			Action: Verify Monitor I is active on the systems from which data is requested, and rerun the program.
12	8	-,-,-,3	**Meaning:** Error - data could not be retrieved. For the specified date and time range, no data could be retrieved by the ERB3XDRS service.
			Action: Check the time range (*start_time* or *end_time*) parameters on the ERB3XDRS service and rerun the program.
12	9	-,-,-,3	**Meaning:** Error - VSAM retrieval errors occurred. For the specified date and time range, no data could be retrieved by the ERB3XDRS service.
			Action: Check the time range (*start_time* or *end_time*) parameters on the ERB3XDRS service and rerun the program.
12	16	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. No data available.
12	17	-,-,-,3	**Meaning:** Error - The Monitor III session is not active on the system specified on the *system_name* parameter of the ERB3XDRS service. If data was requested from all systems in the sysplex, no Monitor III session was found active in the sysplex.
			Action: Start Monitor III on the system(s) for which Monitor II data was requested. Check the system name parameter passed to the ERB3XDRS service.
12	18	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Preallocated data sets unusable (detected at start of retrieval).
12	19	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Preallocated data sets unusable (detected during data retrieval).
12	20	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Too many reporters tried to get data from the in-storage buffer.
12	21	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Retrieval from in-storage buffer failed.
12	22	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. No data in the in-storage buffer.
12	23	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Not enough storage available to copy the requested data from the in-storage buffer.
12	25	-,-,2,-	**Meaning:** Error - SRM STCPS facility not available. The system resource manager (SRM) Store Channel Path Status (STCPS) facility is not available.
12	26	-,-,2,-	**Meaning:** Error - System in WLM GOAL mode. The system is in MVS Workload Manager (WLM) GOAL mode. Therefore, the Monitor II domain activity or transaction activity data (record type 79, subtypes 8 or 10) can not be gathered.
12	27	-,-,2,-	**Meaning:** Error - Transaction data not available. Therefore, the Monitor II transaction activity data (record type 79, subtype 8) cannot be returned.
12	30	-,-,2,3	**Meaning:** Error - Timeout. Due to a timeout situation, ERB2XDGS or ERB3XDRS could not return the requested information.
			Action: Request a smaller amount of information on one call of the ERB2XDGS or ERB3XDRS service.
12	36	Q,-,-,-	**Meaning:** Error - no data returned by ERBDSQRY. No SMF data was found in the sysplex matching the specification provided by the *smf_start_time*, *smf_end_time*, *smf_record_type_info*, *smf_record_type_list*, *smf_system_name_info*, and *smf_system_name_list* parameters of the ERBDSQRY service.
			Action: Check the parameter specifications.

Table 2-7. RMF Sysplex Data Services Return and Reason Codes (SMF Services) (continued)

Return Code	Reason Code	Service Routine	Meaning
			Action
12	37	Q,R,-,-	**Meaning:** Error - All RMF Data Buffers for SMF data are inactive on the systems specified on the *smf_system_name_info* and *smf_system_name_list* parameters of the ERBDSQRY service. For ERBDSREC, an attempt was made to request SMF records from a system on which the RMF data buffer is inactive.
			Action: Start RMF Data Buffer on one or more systems in the sysplex. Check the list of system names passed to the ERBDSQRY service.
12	70	-,-,2,3	**Meaning:** Error - answer area too small. The answer area provided by the calling program was too small for the service to return all the requested information. The variable *answer_area_length* area contains the length of the answer you should have provided for this ERB2XDGS or ERB3XDRS request.
			Action: Provide an answer area large enough to contain all the requested information.
16	0	-,-,-,-	**Meaning:** Reserved for RMF internal use.
			Action: Not applicable.
16	41	Q,-,-,-	**Meaning:** Severe error - The calling program specified an invalid value for the request type (*request_type*). parameter for ERBDSQRY.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	42	Q,-,-,3	**Meaning:** Severe error - The calling program specified an invalid value for the interval/range start or end time (*start_time* or *end_time*) or parameter (YYYYMMDDHHMMSS) on the ERBDSQRY ERB3XDRS service. This includes wrong-formatted parameters and out-of-range or invalid dates.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	43	Q,-,-,-	**Meaning:** Severe error - The calling program specified an invalid value for the SMF record type (*smf_record_type_info*) parameter (INCLUDE/EXCLUDE/ALL) of the ERBDSQRY service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	44	Q,-,-,-	**Meaning:** Severe error - The calling program specified an invalid value for the SMF system name (*smf_system_name_info*) parameter (INCLUDE/EXCLUDE/ALL) of the ERBDSQRY service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	45	-,-,-,3	**Meaning:** Severe error - The calling program specified an invalid value for the data format (*df_ssos* or *df_comp*) subparameters (YES/NO) of the ERB3XDRS service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	46	-,-,2,-	**Meaning:** Severe error - A bad SMF record type or subtype (*rty* or *sty*) was specified for the ERB2XDGS service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	52	-,-,-,3	**Meaning:** Severe error - The calling program specified range start and end times with a difference greater than 9999 seconds in the (*start_time* and *end_time*) parameters of the ERB3XDRS service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	53	Q,-,-,-	**Meaning:** Severe error - An invalid SMF record type or subtype was specified in the record type list (*smf_record_type_list*) for the ERBDSQRY service. Either the length of the list was negative, or a record type was out of the range of 0 to 255.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.

Table 2-7. RMF Sysplex Data Services Return and Reason Codes (SMF Services) (continued)

Return Code	Reason Code	Service Routine	Meaning
			Action
16	54	Q,-,-,-	**Meaning:** Severe error - An invalid SMF system name was specified in the system name list (*smf_system_name_list*) for the ERBDSQRY service, or the length of the list was negative.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	55	Q,-,-,3	**Meaning:** Severe error - An invalid data time interval (*start_time* or *end_time*) was specified for the ERBDSQRY or ERB3XDRS service, i.e. the start time is greater than or equal to the end time.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	56	Q,-,-,-	**Meaning:** Severe error - An empty SMF record type and subtype list (*smf_record_type_list* and *smf_record_type_info* = INCLUDE) was specified for the ERBDSQRY service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	57	Q,-,-,-	**Meaning:** Severe error - An empty SMF system name list (*smf_system_name_list* and *smf_system_name_info* = INCLUDE) was specified for the ERBDSQRY service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	58	-,R,-,-	**Meaning:** Severe error - An empty record token list (*rmf_record_token_list*) was specified for the ERBDSREC service.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	60	Q,R,2,3	**Meaning:** Severe error - RMF could not access one or more of the parameters.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	61	Q,R,2,3	**Meaning:** Severe error - RMF could not access the answer area via the specified ALET (*answer_area_alet*).
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	70	Q,R,2,3	**Meaning:** Severe error - The answer area provided by the calling program (*answer_area_addr* and *answer_area_length*) header was too small to contain even the information.
			Action: Examine your program to locate the CALL that caused the error condition. Correct the statements that are wrong, and rerun your program.
16	71	Q,R,-,-	**Meaning:** Severe error - The requested storage could not be allocated.
			Action: Increase the size of the region where the calling program is running.
16	80	Q,R,-,-	**Meaning:** Severe error - The user is not authorized to call the RMF sysplex data services for SMF data (ERBDSQRY and ERBDSREC).
			Action: Contact your local security administrator.
16	81	Q,R,2,3	**Meaning:** Severe error - The calling program is not in task mode.
			Action: Rerun your program in the correct mode.
16	82	Q,R,2,3	**Meaning:** Severe error - The calling program is not enabled.
			Action: Rerun your program in the correct mode.
16	83	Q,R,2,3	**Meaning:** Severe error - The calling program is not unlocked.
			Action: Rerun your program in the correct mode.
16	90	Q,R,2,3	**Meaning:** Severe error - RMF encountered a severe error when calling the service routine. This may be caused by a terminating RMF address space.
			Action: Restart RMF and rerun your program.

Table 2-7. RMF Sysplex Data Services Return and Reason Codes (SMF Services) (continued)

Return Code	Reason Code	Service Routine	Meaning Action
16	91	-,-,2,3	**Meaning:** Severe error - RMF encountered a severe error when loading the service exit routine. The routine was not found.
			Action: Ensure the exit routine is properly installed on all systems the request is directed to. Rerun your program.
16	92	-,-,2,3	**Meaning:** Severe error - RMF recognized a severe error when executing the service exit routine. The exit completion code is provided in the answer area returned by the service.
			Action: Correct the exit routine problems and rerun your program.
20	0	Q,R,2,3	**Meaning:** Unrecoverable error - An unrecoverable RMF error was encountered during the processing of the requested service. This situation is normally accompanied by error messages sent to the system console and/or a dump.
			Action: Notify your system programmer.

End of Programming Interface information

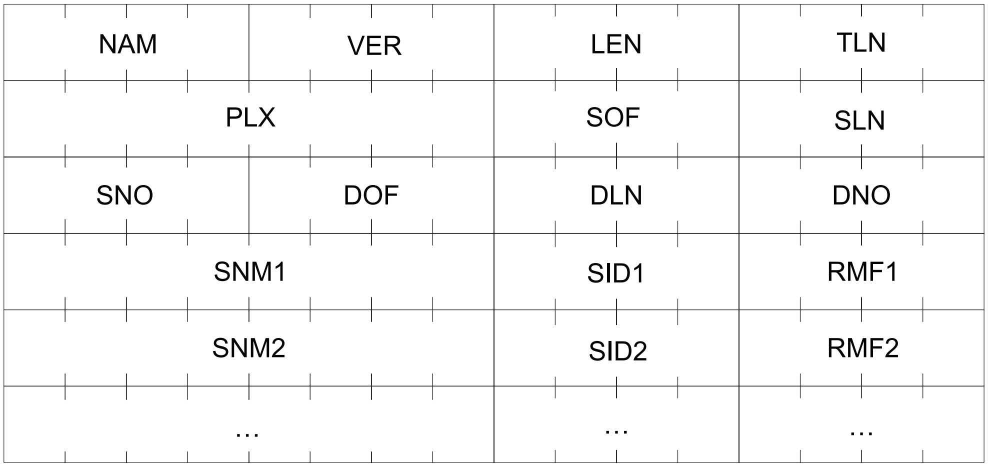
Layout of RMF callable services answer area

Programming Interface information

When ERBDSQRY, ERBDSREC, ERB2XGDS or ERB3XDRS complete successfully and return control to your program, the answer area contains a common header and one or more data sections.

Layout of common answer area header

The layout for the common callable service answer area header is:



NAM	VER	LEN	TLN
PLX		SOF	SLN
SNO	DOF	DLN	DNO
SNM1		SID1	RMF1
SNM2		SID2	RMF2
…		…	…

where:

NAM Four-character acronym of the common header as follows:
- ’DSQA’ for ERBDSQRY
- ’DSRA’ for ERBDSREC
- ’XDGH’ for ERB2XDGS
- ’XDRH’ for ERB3XDRS

VER Version of the common header (initially set to 1).

LEN Total length of the returned data.

TLN Total length of the answer area needed to contain all the requested data.

PLX Name of the sysplex on which the calling application is running.

SOF Offset from the header to the first system list entry SNM.

SLN Length of one system list entry (SNM,SID,RMF).

SNO Number of system list entries (SNM,SID,RMF).

DOF Offset from the header to the first data section. For the detailed layout, refer to the individual data section explanations.

DLN Length of one data section. For a variable length data section, this field is zero. In this case, the length is stored in the individual data section header.

DNO Number of returned data sections.

system list
contains one entry per system in the sysplex:

SNMn
8-character system name

SIDn 4-character SMF system ID. If RMF is not active on this system, this field contains hex zeros.

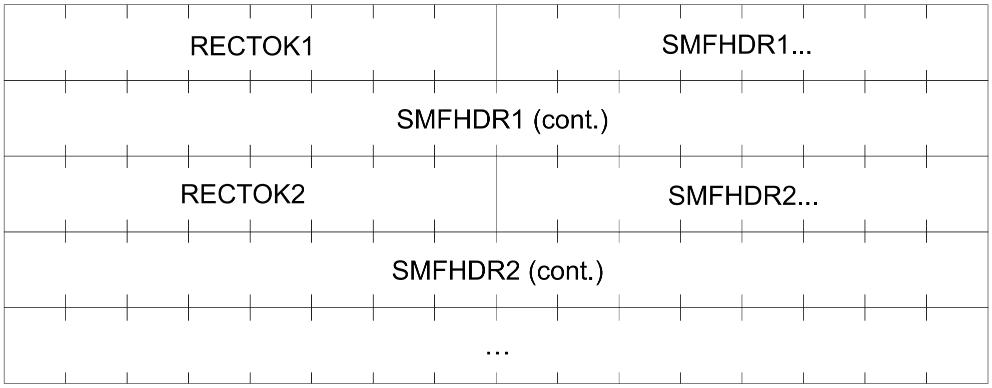
RMFn 32-bit RMF status indicator, in which:

- Bit 0 (high-order bit) indicates the status of the RMF address space on this system (’1’B = active)
- Bit 1 indicates the status of the RMF Data Buffer for SMF data on this system (’1’B = active)
- Bit 2 indicates the status of the RMF Monitor III address space on this system (’1’B = active)
- Bits 3 to 31 are reserved

ERBDSQRY data section layout

When ERBDSQRY completes successfully and returns control to your program, the answer area contains the common header plus one directory entry for each SMF record. The directory entry contains a record token created by ERBDSQRY, which may be used for a subsequent call to ERBDSREC to request the actual SMF record itself, and the SMF record header.

The complete layout for the answer area directory entry for *request_type* = **SMF** is:



Answer area

where:

RECTOKENn Record token provided by ERBDSQRY to be used on subsequent calls to ERBDSREC.

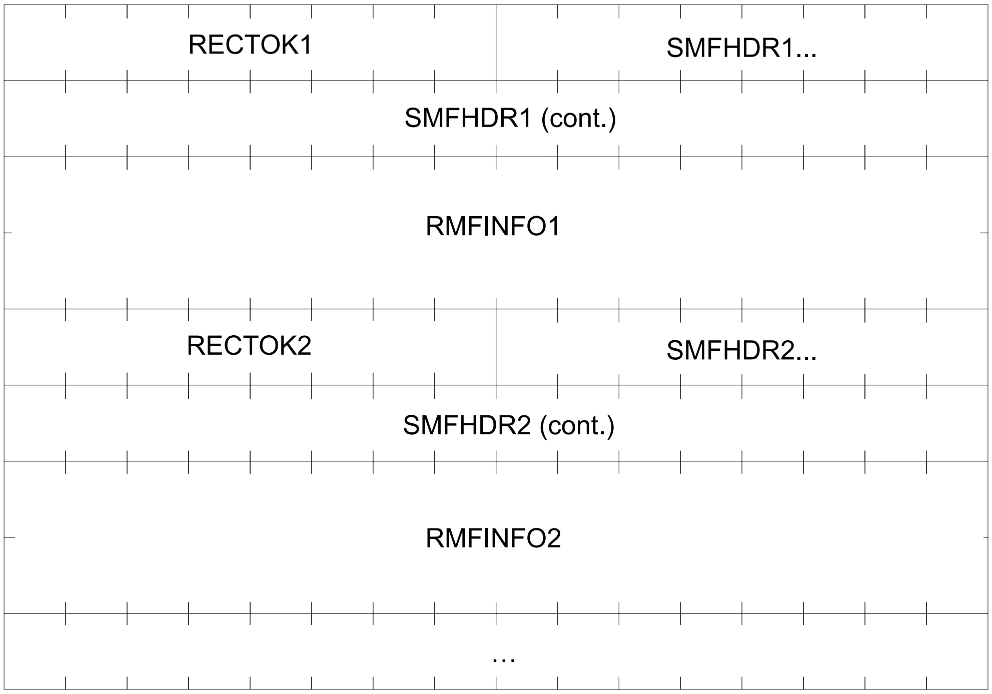
SMFHDRn SMF record header (24 bytes) as described in *z/OS MVS System Management Facilities (SMF)*. For SMF record types without subtypes, which have a header only 18 bytes long, bytes 19 to 24 contain hex zeros.

Name	Length	Format	Description.
SMFxxLEN	2	Integer	SMF record length
SMFxxSEG	2	Integer	SMF segment descriptor
SMFxxFLG	1	Binary	SMF system indicator
SMFxxRTY	1	Integer	SMF record type
SMFxxTME	4	Integer	SMF record time (1/100 sec)
SMFxxDTE	4	0CYYDDDF	SMF record date
SMFxxSID	4	Char	SMF system id
SMFxxSSI	4	Char	SMF subsystem id
SMFxxSTY	2	Integer	SMF record subtype

For *request_type* = **SMF**, the directory entries are sorted by:

1. **SMFxxDTE**: SMF record date
2. **SMFxxTME**: SMF record time
3. **SMFxxRTY**: SMF record type
4. **SMFxxSTY**: SMF record subtype
5. **SMFxxSID**: SMF record system ID

For *request_type* = **RMF** only, each directory entry contains **additional** information from the RMF product section of the SMF record. The layout for *request_type* = *RMF* is:



where:

RMFINFOn For *request_type* = **RMF**, this field contains 32 bytes of additional information from the RMF product section of the SMF record:

Name	**Length**	**Format**	**Description.**
SMFxxDAT	4	0CYYDDDF	RMF actual interval start date
SMFxxIST	4	0HHMMSSF	RMF actual interval start time
SMFxxINT	4	MMSSTTTF	RMF actual interval length
SMFxxOIL	2	Integer	RMF projected interval length (seconds)
SMFxxSYN	2	Integer	RMF synchronization value (seconds)
SMFxxLGO	8	(STCK)	RMF offset GMT to local time
SMFxxGIE	8	(STCK)	RMF projected interval end (GMT)

For *request_type* = **RMF**, the directory entries are sorted by:

1. **SMFxxDAT**: RMF interval start date
2. **SMFxxIST**: RMF interval start time
3. **SMFxxRTY**: SMF record type
4. **SMFxxSTY**: SMF record subtype
5. **SMFxxSID**: SMF record system ID

ERBDSREC data section layout

When ERBDSREC returns control to your program after the service was completed successfully, the answer area contains the common header and one entry for each requested SMF record. The entries appear in the order of the request, which is identical to the order of the tokens in the record token list. The entry for each record contains a data header, which is provided by ERBDSREC, and the SMF record itself.

Answer area

The complete layout of the data section is as follows:

<table>
<tr><td>RL1</td><td>RH1</td><td>RC1</td><td>*</td></tr>
<tr><td colspan="2">RECTOK1</td><td colspan="2">SMFRECORD1.</td></tr>
<tr><td colspan="4">SMFRECORD1 (cont.)</td></tr>
<tr><td>RL2</td><td>RH2</td><td>RC2</td><td>*</td></tr>
<tr><td colspan="2">RECTOK2</td><td colspan="2">SMFRECORD2...</td></tr>
<tr><td colspan="4">SMFRECORD2 (cont.)</td></tr>
<tr><td colspan="4">...</td></tr>
</table>

where:

RLn Length of this SMF record data entry, including the data header

RHn Length of this SMF record data header

RCn Return code for the request of this SMF record:

- **0** Data returned. SMF record data follows this data header
- **4** Data not returned. Timeout occurred before the record was received from the remote system
- **8** Data not returned. The record token does not correspond to an existing SMF record in the sysplex

RECTOKn Record token for this SMF record (copied from input parameter)

SMFRECORDn
SMF record

ERB2XDGS data section layout

When ERB2XDGS returns control to your program after the service was completed successfully, the answer area contains the common header and one or more data sections. Each data section contains a data header followed by the Monitor II data itself.

The layout of the data header is

Answer area

DEL	HDL	RTN	RSN
CPU	PRT	DRC	
…	SRM	SID	

where:

DEL Length of this data section

HDL Length of this data header

RTN Data Retrieval return code

RSN Data Retrieval reason code

CPU System CPU Utilization (if Monitor I CPU gathering is not active, this field has the value ′-1′)

PRT System Paging Rate

DRC Data Reduction exit completion code, if the exit ended abnormally. The completion is in the format TCCCRRRRRRRR, where:
- T is ′S′ or ′U′ for a system or user completion code, respectively
- CCC is the hexadecimal completion code. The highest possible user completion code is x′FFF′.
- RRRRRRRR is the hexadecimal reason code associated with the completion code.

SRM MVS view of CPU utilization if Monitor I CPU gathering is active, otherwise the SRM view of the CPU utilization (CCVUTILP).

SID SMF system ID.

Each data section contains the data header described above, followed by the data provided by the data reduction exit routine.

ERB3XDRS data section layout

When ERB3XDRS returns control to your program after the service has completed successfully, the answer area contains the common header and one or more data sections. Each data section contains a data header followed by the Monitor III data itself. The layout of the data section is as follows:

- One or more set-of-samples. The layout of the uncompressed set-of-samples is described in "Data gatherer sample structure" on page 5-3.

The layout of the data header is

Answer area

<table>
<tr><td colspan="4">DEL</td><td colspan="4">HDL</td><td colspan="4">RTN</td><td colspan="4">RSN</td></tr>
<tr><td colspan="3">DGV</td><td colspan="5">*</td><td colspan="4">DGS</td><td colspan="4">MNT</td></tr>
<tr><td colspan="4">SAM</td><td colspan="4">RNG</td><td colspan="8">BEG</td></tr>
<tr><td colspan="6">…</td><td colspan="10">END</td></tr>
<tr><td colspan="4">…</td><td colspan="12">DRC</td></tr>
<tr><td colspan="8">DSG</td><td colspan="8">DEG</td></tr>
<tr><td colspan="8">DIT</td><td colspan="8">DFA</td></tr>
<tr><td colspan="8">DLA</td><td colspan="8">…</td></tr>
</table>

where:

DEL Length of this data section

HDL Length of this data header

RTN Data Retrieval return code

RSN Data Retrieval reason code

DGV Data gatherer version in the format ’VRM’.

DGS System name of the system on which the data gatherer is running

MNT Data gatherer MINTIME option

SAM Actual number of samples in the returned data

RNG Actual range length in seconds

BEG Actual range start time in the format YYYYMMDDHHMMSS.

END Actual range end time in the format YYYYMMDDHHMMSS.

DRC Data Reduction exit completion code, if the exit ended abnormally The completion code is in the format TCCCRRRRRRRR, where:
- T is ’S’ or ’U’ for a system or a user completion code, respectively
- CCC is the hexadecimal completion code
- RRRRRRRR is the hexadecimal reason code associated with the completion code

The following fields will be filled with Monitor III data statistics for certain warning and error conditions.

For return code 8 or 12 and reason code 8 or 9:

DSG Start time of a time gap in the Monitor III data in store clock format

DEG End time of a time gap in the Monitor III data in store clock format

For return code 8 or 12 and reason code 15:

DIT IPL time of the system in store clock format

For return code 12 and reason code 16:

DFA Start time of the Monitor III data that is available for reporting on this system in store clock format

DLA End time of the Monitor III data that is available for reporting on this system in store clock format

* Reserved

Note: The data header length field contains 120 instead of 80 if the additional data statistics are present. If the systems in the sysplex have a different RMF service level, both data header formats may appear in the same ERB3XDRS answer area.

Each data section contains the data header described above, followed by the data provided by the data reduction exit routine.

End of Programming Interface information

Answer area

Chapter 3. Using LDAP to access performance data

Accessing Performance Data from a Standard Interface

This chapter describes how you can use the LDAP protocol to access RMF performance data from application programs. It informs you how RMF data is stored within LDAP and how to access it from C and Java applications as well as from a web browser which suppports the LDAP protocol. A sample Java program and several sample URLs are also provided.

The following topics are discussed:

- How RMF processes data from LDAP
- How to set up RMF LDAP integration
- How to code LDAP access methods

Introduction to LDAP

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines methods for accessing and updating information in a directory.

A directory is a listing of information about objects arranged in some order that gives details about each object. Common examples are a city telephone directory and a library card catalog. For a telephone directory, the objects listed are people; the names are arranged alphabetically, and the details given about each person are address and telephone number. Books in a library card catalog are ordered by author or by title, and information such as the ISBN number of the book and other publication information is given.

Normally, with LDAP, new entries can be added, existing entries can be altered or deleted. However, using LDAP to retrieve RMF performance data, these functions of updating information do not make sense and therefore are not supported.

LDAP is an Internet Protocol Standard based on the TCP/IP protocol which makes directory information accessible. The LDAP directory is represented hierarchically in a Directory Information Tree (DIT). The nodes of this tree are called entries. Every entry is an instance of an object class. An object class is a general description, sometimes called a template, of an object as opposed to the description of a particular object. The RMF DIT of objects is depicted in Figure 3-3 on page 3-8.

Each entry contains one or more attributes that describe the entry. Each attribute has a type and a value with the value either being single valued or multi valued. In RMF, every entry has two required attributes: a ***name*** and an ***objectclass***. The second attribute specifies the object class of which the entry is an instance.

Each entry has a relative distinguished name (RDN) which identifies the entry uniquely among its siblings (not necessarily in the whole tree). An entry is described by one or more attribute-value pairs. The distinguished name (DN) of an entry is the sequence of the RDNs starting from the entry itself and ending with the RDN of the root entry. The DN identifies an entry uniquely within the whole DIT. You will find several examples of distinguished names for RMF entries in "The LDAP directory structure" on page 3-7.

In z/OS, LDAP services are available through the z/OS Security Server LDAP Server.

LDAP Documentation

For detailed information about LDAP refer to *z/OS Security Server LDAP Server Administration and Use*, to *z/OS Security Server LDAP Client Programming*, or to the redbook *Understanding LDAP* (SG24-4986) from http://www.redbooks.ibm.com.

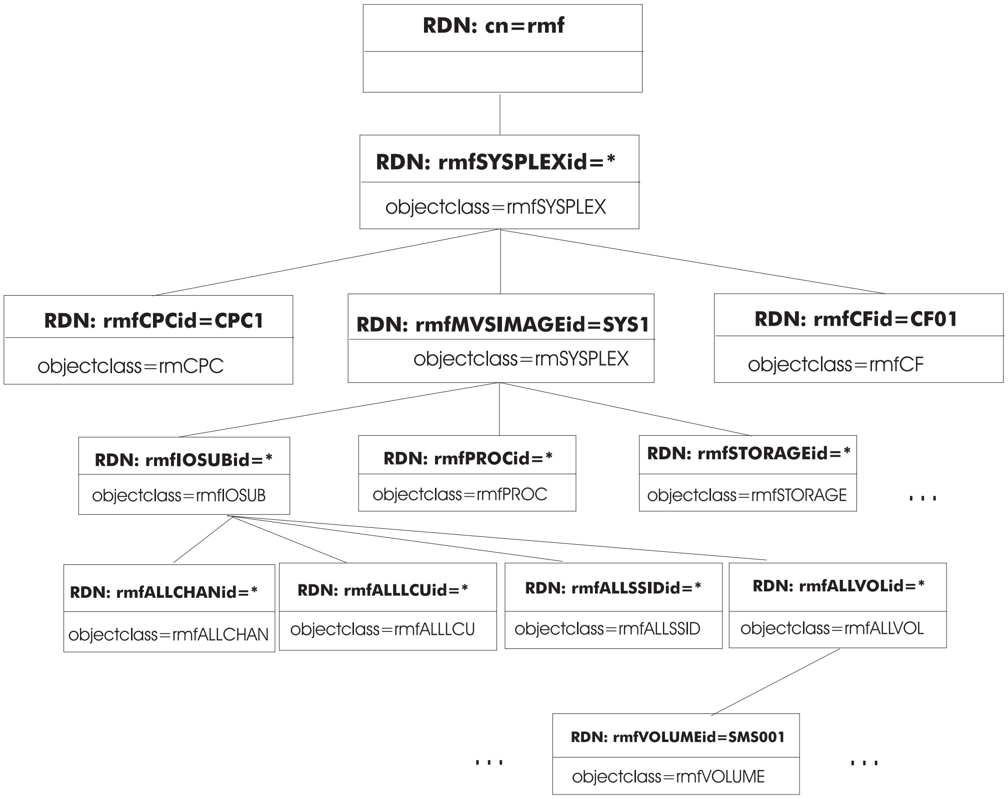


Figure 3-1. Sample DIT Portion of the RMF LDAP Performance Data

The z/OS LDAP Server delegates the task of storing the DIT to several so-called backends. These DITs can be accessed by a client via the LDAP protocol. Each backend is responsible for carrying out the client's requests. The DN of the root entry of a backend is called a ***suffix*** and is part of the DN of any entry. When the z/OS LDAP Server receives a request concerning an entry, it identifies the suffix within the DN of that entry in order to determine which backend is responsible for the request.

RMF supports the z/OS LDAP Server by providing a backend which makes performance data from the RMF Distributed Data Server (DDS) accessible via the LDAP protocol.

How to access RMF performance data from LDAP

Using LDAP, you can access RMF performance data from application programs. This allows other performance and systems management components easy access to performance data within an z/OS environment.

The LDAP directory is like a database, containing the RMF performance data in a hierarchical structure. LDAP offers well-defined access methods to this data.

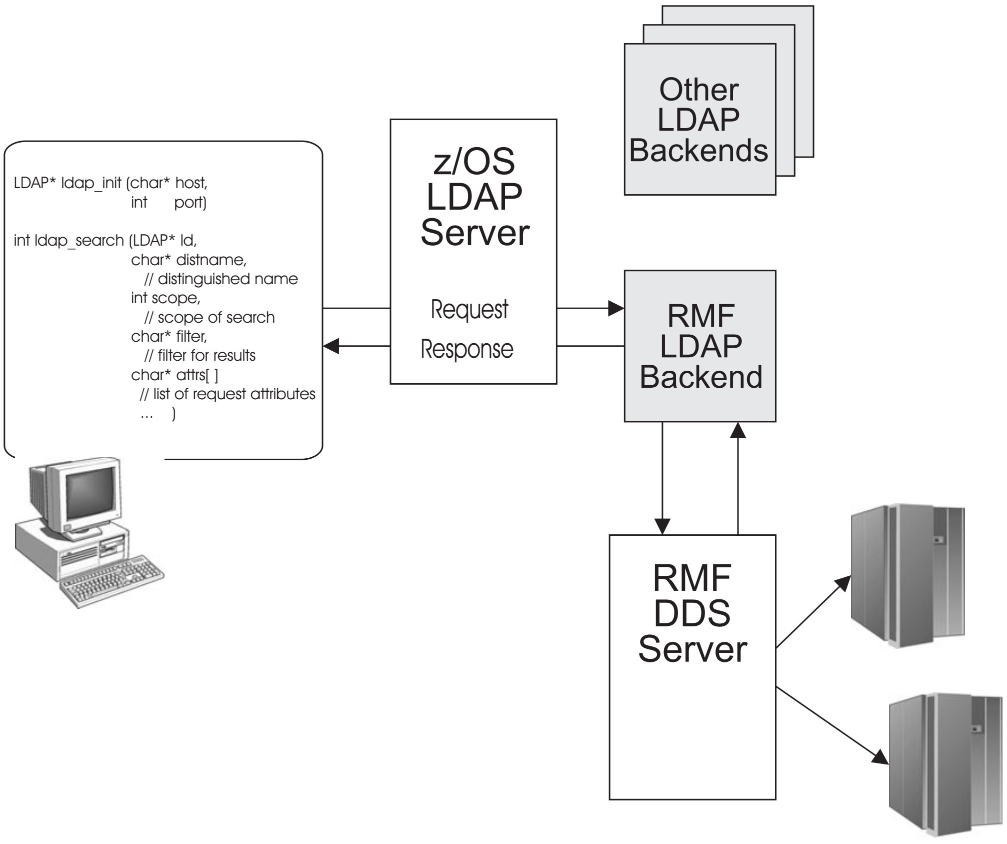
The process is as follows (see also Figure 3-2 on page 3-5):

- Your application program executes an LDAP access method, actually a request to access performance data.
- This request is sent to the z/OS LDAP Server, which promotes this request to the RMF LDAP backend.
- The RMF LDAP backend contacts the RMF Distributed Data Server (DDS) and makes the requested data accessible in LDAP format. More than 600 z/OS performance metrics are accessible this way!
- On the same way as the request reached the DDS, the response with the requested data is propagated back to your application program.

LDAP search requests for the RMF backend can be issued from any standard LDAP client. Though the main purpose of LDAP integration into RMF is to access performance data from application programs, you can also issue LDAP requests from any web browser, which supports the LDAP protocol. This can be useful for a quick lookup of a special metric.

In the subsequent sections, you find information required for getting started in LDAP programming for C and JAVA.

Additional samples of C and Java application programs and the URL syntax of LDAP requests are shown in the previously mentioned redbook *Understanding LDAP*.



```
LDAP* ldap_init (char* host,
                 int      port)

int ldap_search (LDAP* ld,
                 char* distname,
                   // distinguished name
                 int scope,
                   // scope of search
                 char* filter,
                   // filter for results
                 char* attrs[ ]
                  // list of request attributes
                  ...    )
```

Figure 3-2. Accessing Performance Data from LDAP

Note: As a prerequisite, the Distributed Data Server (DDS) has to be active in order to provide the performance data source for the RMF LDAP backend.

How to set up RMF LDAP integration

To set up the communication between the RMF LDAP backend and the z/OS LDAP Server, you need to perform configuration steps on both sides, as described in the following sections.

How to set up the z/OS LDAP Server

The z/OS LDAP Server, shown in Figure 3-2, is able to communicate with one or more backends by means of a well defined interface. When starting the z/OS LDAP Server, you must specify all the backends with which this server should communicate. This is done in the z/OS LDAP Server configuration file **SLAPD.CONF**. So you need to append the configuration information for the RMF LDAP backend to this file. Sample RMF backend information is presented in "How to configure the RMF LDAP backend".

A sample JCL to start the z/OS LDAP Server is provided in **SYS1.SGLDSAMP(LDAPSRV)**.

How to configure the RMF LDAP backend

This section presents a sample of a configuration file for the RMF backend. You can find it on your host in data set **SYS1.SAMPLIB(ERBSLAPD)**. This file must be edited according to your environment. The RMF LDAP backend function

ERB6LBCK needs to be declared in the database section of this file. Thus, the RMF LDAP backend will be automatically loaded during the z/OS LDAP Server initialization.

The z/OS LDAP Server identifies the backend that is responsible for a certain request with the help of the suffix parameter, which is specified with the RMF backend configuration. Thus, among others, you have to specify the following entries in the database section of the RMF configuration information to set up LDAP integration for RMF:

```
database rmf ERB6LBCK
suffix "cn=rmf"
```

So the RMF LDAP backend will serve all requests that arrive with the suffix ***cn=rmf***, as described in “How to code LDAP access methods” on page 3-9.

```
#######################################################################
# rmf database definitions
#######################################################################

database rmf ERB6LBCK

# The following options must be filled in with appropriate values
# for your RMF setup, prior to attempting to run with the RMF backend.

# ---------------------------------------
#
# suffix <toplevelname>
#
# Default Value: none
#
# Example:
#   suffix "cn=rmf"
#
# NOTES:
#   This option is REQUIRED when using the RMF database.
#
# ---------------------------------------
# suffix "cn=rmf"


# ---------------------------------------
#
# rmfURL  <httpURL>
#
# Default Value: none
#
# Example:
#   rmfURL http://sysplex.yourcompany.com:8803
#
# NOTES:
#
#   The URL must identify a valid hostname or IP address, where the
#   RMF Distributed Data Server (DDS) is listening for requests at
#   the specified port.
#   The host running the RMF LDAP backend must be specified in the
#   HTTP_NOAUTH statement of the parmlib member for the RMF DDS Server
#   Example: HTTP_NOAUTH(ldap.boeblingen.ibm.com)
#
# ---------------------------------------
# rmfURL http://sysplex.yourcompany.com:8803

# ---------------------------------------
#
# rmfRoot <rootresource>
#
```

```
# Default Value: *,*,SYSPLEX,1
#
# Example:
#   rmfRoot SYS1,*,IOSUB,1
#
# NOTES:
#
#   The root resource is a comma seperated string with four qualifiers:
#   HLQ(system or coupling facility name), resource name, resource type
#   and the indication, if the specified resource has children or not
#   (1=yes, 0=no).
#   This allows to retrieve configuration and performance data only for
#   a subset of a sysplex, e.g. the I/O subsystem of a specific system.
#
# ---------------------------------------

# ---------------------------------------
#
# rmfSecurity <none|racf>
#
# Default Value: none
#
# Example:
#   rmfSecurity racf
#
# NOTES:
#
#   If rmfSecurity racf is specified, a bind request to the rmf backend
#   is checked against the LDAP racf backend for a valid TSO userid and
#   password. As a perequisite, the racf backend has to be configured
#   and must be active.
#
# ---------------------------------------
```

The LDAP directory structure

Before you can use an application program to access data from LDAP, you need to understand the LDAP directory structure.

In LDAP, directory entries are arranged in a hierarchical tree-like structure that reflects, for example, the structure of a sysplex. A sysplex consists of several systems (images), and every image consists of a processor, an I/O subsystem, storage and other components.

All entries in the LDAP directory belong to ***object classes***. The object classes used by the RMF LDAP backend are structured in the following hierarchy:

```
RMF Object Classes for LDAP        RMF Resource Type


rmfSYSPLEX                         ---> SYSPLEX
  rmfCPC                           ---> CPC
  rmfMVSIMAGE                      ---> MVS_IMAGE
    rmfIOSUB                       ---> I/O_SUBSYSTEM
      rmfALLCHAN                   ---> ALL_CHANNELS
        rmfCHANNEL                 ---> CHANNEL_PATH
      rmfALLLCU                    ---> All_LCUS
        rmfLCU                     ---> LOGICAL_CONTROL_UNIT
      rmfALLSSID                   ---> ALL_SSIDS
        rmfSSID                    ---> SSID
      rmfALLVOL                    ---> ALL_VOLUMES
        rmfVOLUME                  ---> VOLUME
    rmfPROC                        ---> PROCESSOR
    rmfSTORAGE                     ---> STORAGE
      rmfCENTRAL                   ---> CENTRAL_STORAGE
        rmfCSA                     ---> CSA
        rmfSQA                     ---> SQA
        rmfECSA                    ---> ECSA
        rmfESQA                    ---> ESQA
      rmfEXPANDED                  ---> EXPANDED_STORAGE
      rmfAUX                       ---> AUXILIARY_STORAGE
    rmfENQUEUE                     ---> ENQUEUE
    rmfOPERATOR                    ---> OPERATOR
    rmfSWSUB                       ---> SW_SUBSYSTEMS
      rmfJES                       ---> JES
      rmfHSM                       ---> HSM
      rmfXCF                       ---> XCF
  rmfCF                            ---> COUPLING_FACILITY
    rmfCFSTRUC                     ---> CF_STRUCTURE
```

Figure 3-3. RMF Object Classes in LDAP

In your application program, you address the performance data by means of a composed string, which is referred to as ***distinguished name***. The distinguished name contains a unique identifier for each object of the hierarchy of the LDAP directory.

Therefore, if you want to access a specific volume (SMS001) of a specific system (SYS1), you use the following distinguished name:

```
rmfVOLUMEid=SMS001,rmfALLVOLid=*,rmfIOSUBid=*,rmfMVSIMAGEid=SYS1,
rmfSYSPLEXid=*,cn=rmf
```

Notes:

1. You begin your distinguished name with the identification of the object, that you want to address and then you climb up the hierarchy as shown in Figure 3-3. Note that you need to append the string ***id*** (in lower case) at the end of an object class to identify a certain entry from that class. If you want to address, for example, an instance of object class *rmfVOLUME*, you specify *rmfVOLUMEid=volumename*.

   Your distinguished name must always contain the suffix ***cn=rmf*** at its end.
2. For most of the object classes, only one single entry can exist within the structure. In these cases, the resource name can be specified by an asterisk. The following object classes can have one or more entries and therefore, you need to specify the resource name explicitly:
   - rmfCPC
   - rmfMVSIMAGE
   - rmfCHANNEL

- rmfLCU
- rmfVOLUME
- rmfCF
- rmfCFSTRUC

LDAP requests can have three different scopes:

base Search and return only the specific entry specified with the distinguished name

one Search and return one level below the entry specified with the distinguished name; this entry itself will not be returned.

sub Search and return all levels below the entry specified with the distinguished name; this entry itself will not be returned.

For example, if you specify in your API request the distinguished name

```
rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,cn=rmf
```

and you want to search one level below this name (specifying scope: **one**), then RMF will return information for the parent image `SYS1` as well as for the children:

```
rmfIOSUB, rmfPROC, rmfSTORAGE, rmfENQUEUE, rmfOPERATOR, rmfSWSUB
```

Other useful examples of using scope **one** would be to search with one request for all volumes (starting from `rmfALLVOL`) or for all coupling facility structures (starting from `rmfCF`).

Notes:

1. The deeper the level that you want to search, the bigger will be the amount of returned data.
2. Depending on the API you use, you either must specify a scope or a certain scope is used as a default. For more information, refer to the description of the API you are using.

How to code LDAP access methods

LDAP provides APIs for some high level languages, for example, Java programs or C programs. So you can write such programs that may contain LDAP access methods, also called ***LDAP requests***, to monitor special performance data and trigger some automation if a defined threshold is reached.

An LDAP request addresses the performance data, actually a performance data object, by means of the distinguished name as described in “The LDAP directory structure” on page 3-7.

The way you code an LDAP request depends on the API that you are using. Figure 3-2 on page 3-5 illustrates an example for a C program and the Java API is demonstrated in “Java sample program” on page 3-13. The LDAP response delivers back the entry that you requested together with the attributes and their values.

Performance data stored in LDAP

This section informs you about the kind of performance data that is stored for entries of each of the object classes listed in “The LDAP directory structure” on page 3-7. The performance data is attached to an entry by means of one or more attribute-value pairs. In your request, you specify the attributes of which you want to know the values. The response then returns these values.

Examples of attributes are the percentage of total utilization of a processor (specified as attribute: rmf8d0460) or the number of active users in a sysplex (rmf8d0620). More than 600 attributes (RMF metrics) can be retrieved by means of LDAP.

The information about available attributes for the RMF object classes and the attribute IDs (metric IDs) is stored in data set **SYS1.SERBPWSV(GPMOCATT)**. Download this data set to your workstation as HTML file. For example in Table 3-1, which is an excerpt from this file as seen from a browser, you can see that the attribute # active users with the attribute ID 0x8D0620 is available for the object class SYSPLEX.

Table 3-1. Object Classes and Attributes

RMF DDS Metric Name	Supported Workscopes	Metric id	List types
SYSPLEX (LDAP short name: SYSPLEX)			
# active users	G,W,S,P,R	0x8D0620	-
# delayed i/o requests	G	0x8D0680	-
# delayed jobs	G,W,S,P,R	0x8D12D0	-
...	...	...	...
COUPLING_FACILITY (LDAP short name: CF)			
# effective logical processors	G	0x8D1FF0	-
# frames available	G	0x8D2000	-
...	...	...	...

Notes:

1. Attributes for RMF performance data are denoted by a four-digit ID, which is preceded by the suffix *rmf8d*. So the metrics (attributes) shown as 0x8Dnnnn in Table 3-1 must be specified as rmf8dnnnn in your LDAP requests.
2. Only those attributes can be used in LDAP requests which show a ***G*** in column ***Supported Workscopes***. ***G*** stands for *global workscope*, as special workscopes like service classes or report classes are not supported.
3. It is reasonable to specify attributes only for object classes to which they are related. For example, the attribute rmf8d0620 (# active users) is supported for a sysplex, but not supported for a volume. If you nevertheless request an unsupported attribute, it will not be contained in the response.
4. The ***List types*** column indicates whether the attribute returns a single value (indicated by dash) or a list-valued counter (indicated by a letter; for example, M will return values for a list of MVS images).

Coding Example:

The number of delayed jobs on an MVS image is denoted as `rmf8d12d0`.

So to get this information for MVS image *SYS1* in the sysplex, you can specify the request in the following ways:

- URL Request:

  ```
  ldap//boesysf:389/rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,cn=rmf?rmf8d12d0
  ```

  Note: In URL syntax, a ***?*** is used as a delimiter between the parts of an LDAP request.

- Java Program:

  ```
  // entry to be searched is specified with its distinguished name:
  String distname = "rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,cn=rmf";
  ...

  SearchControls constraints = new SearchControls();
  ...

  String metric;
  metric="rmf8d12d0";                         // metric to look up
  String attrList[] = {metric};               // list of returned attributes
  constraints.setReturningAttributes(attrList);

  env = new Properties();
  env.put("java.naming.factory.initial",
          "com.sun.jndi.ldap.LdapCtxFactory");

  NamingEnumeration results = new InitialDirContext(env).
                                  search(distname,
                            "(objectclass=*)", // filter
                                  constraints );
  ```

Different Types of Counters:

You can retrieve the following performance data for your sysplex:

- Total number of users in the sysplex (`rmf8d0d50`), which is a single counter
- Number of users by system in the sysplex (`rmf8d0d60`), which is a list-valued counter

You specify the following information in your request:

```
distinguished name:    rmfSYSPLEXid=*,cn=rmf
attributes:            rmf8d0d50,rmf8d0d60
```

So depending on how your application program displays the response, the output from our sample LDAP request could look like follows:

```
CN              *
Object Class    rmfSYSPLEX
rmf8d0d50       801                     # users
rmf8d0d60       SYS1,214                # users by MVS image
                SYS2,167
                SYS3,123
                SYS4,149
                SYS5,148
```

If you do not specify any attributes, the following happens:

- For **scope=base**: If only a distinguished name but no attributes are specified with an LDAP request, the RMF LDAP backend will return a response containing the entry name, its object class and all available attribute-value pairs.
- For **scope=one** or **scope=sub**: The request will return only the common name (CN) and the object class.

LDAP request syntax

To summarize and enhance the information of the previous sections, this section describes the general components of LDAP requests. Sample requests are provided in Table 3-2 on page 3-13.

A normal LDAP request consists of the following parts that you need to consider in your application program (in a URL, these parts are separated by the **?** delimiter):

- **distinguished name**
- **list of attributes**
- **scope**

For example, in a C program, you could compose an LDAP request from the following parameters:

```
char* distname,  // distinguished name
int   scope,     // scope
char* filter,    // filter
char* attrs[],   // list of requested attributes

ld = ldap_init(ldaphost, ldapport);

ldap_search(ld, distname, scope, filter, attrs, attrsonly);
```

The single parts are introduced in more detail:

distinguished name (required)

Examples:

```
rmfSYSPLEXid=*,cn=rmf
rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,cn=rmf
rmfPROCid=*,rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,cn=rmf
```

attribute(s) (optional)

Examples:

```
rmf8d0d50 ---> returns the number of users for the specified sysplex
rmf8d1530 ---> returns the number of using jobs for a sysplex
rmf8d12d0 ---> returns the number of delayed jobs for an MVS image
```

scope (optional)

This part can have the values **base**, **one** or **sub**, as described in “The LDAP directory structure” on page 3-7.

Sample requests and sample program

This section contains a table with several LDAP requests that demonstrate some search facilities. For test purposes or for quick look up, you can issue the search requests for the RMF backend as URL from a web browser, which supports the LDAP protocol. Begin your URLs with:

```
ldap://ldapserv/distinguished_name...
```

where `ldapserv` is the name and port of your z/OS LDAP Server.

How to imbed LDAP requests into a Java application program is shown in "Java sample program".

Table 3-2. LDAP Sample Requests

Request	Explanation
DN: rmfSYSPLEXid=* Attributes: rmf8d0d50 Scope: base URL: /rmfSYSPLEXid=*?rmf8d0d50?base	Returns the sysplex entry and gets the attribute ″number of users″
DN: rmfPROCid=*,rmfMVSIMAGEid=SYS1 Attributes: rmf8d0460 Scope: base URL: /rmfPROCid=*,rmfMVSIMAGEid=SYS1?rmf8d0460?base	Returns the processor entry of MVS image SYS1 and gets the attribute ″% total utilization″
DN: rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=* Attributes: not specified Scope: base URL: /rmfMVSIMAGEid=SYS1,rmfSYSPLEXid=*,?base	Returns the MVS image SYS1 and gets all required attributes for this system
DN: rmfSYSPLEXid=* Attributes: rmf8d0550 Scope: one URL: /rmfSYSPLEXid=*,?rmf8d0550?one	Returns sysplex and direct subentries and gets the attribute ″% workflow″

Java sample program

A Java sample program which can be used as a base for your performance monitoring applications that exploit LDAP, is provided in data set **SYS1.SAMPLIB(ERBJLDAP)**. The prerequisite to compile and run this sample is ***IBM JDK 1.3*** or later.

Invoke this program with any SYSPLEX performance metric that you want to monitor as an input parameter. As this parameter is optional, the program returns the `% active time by volume` (`rmf8d0020`) for the object class `SYSPLEX` as a default.

Note: As in this sample, the distinguished name is hard-coded as `"rmfSYSPLEXid=*,cn=rmf"`, you can provide only attributes of object class `SYSPLEX` as input parameter. In the sample code, you find comments on how to change the program logic to vary also the distinguished name.

Also, the scope of the search is hard-coded to **base**.

When downloading this program, you must rename the member name **ERBJLDAP** to the file name that matches the java class name. For this purpose, use the following command in an ftp-session connected to your host, which downloads and renames in one step:

```
ftp> get ERBJLDAP RmfLdapSearch.java
```

Compile the program using the following command:

```
javac RmfLdapSearch.java
```

Run the program using the following command:

```
java RmfLdapSearch <ldap_server> <ldap_port> [<metric>]
```

with:

ldap_server
TCP/IP name of your LDAP server

ldap_port
TCP/IP port of your LDAP server

metric Attribute ID of the metric you are interested in. This parameter is optional. If specified, you can use only attributes supported for object class SYSPLEX.

The LDAP request is issued in the searchRequest(attrList) method. You can call it in your programs as follows:

```
String metric = "rmf8d0020"; // or any other metric you'd like
String attrList[] = {metric};
NamingEnumeration results = rmfLdap.searchRequest( attrList );
```

In our example, the LDAP results are processed in the printResult method. You may adapt this processing to your requirements.

```
  public void printResults( NamingEnumeration results ) {
  if ( results.hasMore() ) {
SearchResult sr = (SearchResult) results.next();
Attributes attrs = sr.getAttributes();
if ( null != attrs ) {
    NamingEnumeration namingEnum = attrs.getAll();
    while ( namingEnum.hasMore() ) {

                      // changes should be made here
 System.out.println( ((Attribute) namingEnum.next())
       .getAll().nextElement() );

    }
}
   }
  }
```

The rest of the program can be used without changes. If case of errors, you can generate detailed LDAP trace output by adding the following line to your program:

```
cv_env.put("com.sun.jndi.ldap.trace.ber", System.err);
```

Chapter 4. Adding Monitor I and Monitor II installation exits

About Writing Installation Exits

This chapter describes:

- How to create Monitor I user exit routines
- How to create Monitor II user reports

Overview

Facilities in RMF allow you to gather and report data relevant to your installation.

During a Monitor I session, installation exits let you sample data at each RMF cycle, collect this data and examine system indicators at each RMF interval, format and write your own SMF records, and format and write your own reports. You can also use the RMF trace facilities to trace the contents of any SQA, fixed CSA, or nucleus field that you require. During a Monitor II session, the data interface service allows you to directly access SMF record data from storage in real time rather than through SMF. The service provides easy access to this data by invoking the module ERBSMFI.

During a Monitor II session, installation exits enable you to gather and report your own data by coding your own data-gathering and data-reporting routines. RMF provides the USER option for a background session and the USER menu item for a display session. To generate one additional report, you replace module ERBGUS99 with your data gatherer and ERBRUS99 with your data reporter. Specifying USER then causes your own report to be generated. Should you want to obtain more than one user report, you must add an entry to the option list or menu list as well as supply a data-gathering and a data-reporting routine. Data gathered for your routine can be reported either during the session or during execution of the Postprocessor.

During a Monitor II TSO/E display session, with TSO/E installed, a user exit enables your installation to verify that a terminal user is authorized to use RMF. See "TSO terminal user authorization" on page 4-27 for an explanation of this user exit.

Monitor I session user reports

Programming Interface information

To gather and report data relevant to your installation during a Monitor I session, RMF provides both the EXITS option and installation exits at various points during Monitor I session processing. When EXITS is specified, you can:

- Initialize for the other user exit routines
- Sample fixed CSA, SQA, or nucleus data at each RMF cycle
- Perform interval processing, for example, reduce sampled data, examine system state indicators, format SMF records to be written to the SMF data set or passed to your report writer
- Write reports during a session
- Handle termination processing for the other installation exits
- Write reports during execution of the Postprocessor.

In addition, you can use the Monitor I session tracing routines to trace the contents of a fixed SQA, CSA, or nucleus field regardless of whether or not EXITS is specified.

Guidelines

Each of the user functions is described in detail in the following sections. The following guidelines apply to Monitor I user exit routines:

- All of the user exit routines must be reenterable.

- All user-written exit routines receive control in 31-bit addressing mode.
- The routines must save registers when they receive control and restore registers when they return control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.
- One input parameter that RMF passes to each user exit routine (except the tracing routine and the Postprocessor user exit) is the address of a two-word area reserved for the use of your routines. Because these words provide a means of communication between your exit routines, their use should be controlled by conventions agreed upon by your installation.
- RMF passes a phase parameter to each user exit routine except the sampler, the tracing routine, and the Postprocessor user exit. This phase parameter indicates which RMF phase is invoking the user exit.

RMF provides dummy routines for all Monitor I session exits that are not used.

Caution

Because all of the user exit routines except ERBMFRUR (the report writer) run in supervisor state with a key of 0, your installation must carefully control their use. Program errors that cause an exit routine to overlay system areas could bring down the system.

Initialization for Monitor I session user exit routines

The initialization user exit is ERBMFIUC. It is called at the start of a Monitor I session and whenever the Monitor I session options are modified. Use this exit to perform any initialization the other installation exits require, such as building a control block structure.

When the exit routine gets control, register 1 points to a three-word address list. The first address points to the two-word area reserved for use by your routines. This same two-word area is passed to all the user exit routines and can be used for communication between them. The second address points to the RMF phase parameter, a full-word field that is always X‘4:’, indicating that the exit is called during Monitor I session initialization. The third address points to a word that is relevant only when you are providing a routine to sample data at each cycle; one of the functions your initialization routine will perform is to put the address of the user sampler in this word. Figure 4-1 illustrates the input parameter structure.

When the initialization routine is entered, the system is in supervisor state, and all interrupts are enabled. ERBMFIUC runs in key 0.

Special initialization procedures are required when your user routines include a sampling routine to sample data at each cycle; see “Sampling Data at Each Cycle.” When you have a user sampler, your initialization routine **must** do the following:

- The user sampling routine must be loaded and page fixed. You must use the PGSER macro to page fix the user sampler routine because the sampler code runs disabled.
- The address of the user sampling routine must be placed in the third input parameter.
- All storage the sampler routine will require must be obtained; this storage must be obtained from SQA (subpool 245).
- The address of the SQA storage obtained must be placed in one of the two user words. The choice depends on the conventions established at your installation.

Mon I initialization

When you have completed the initialization required by all the installation exits, return control by branching on register 14.

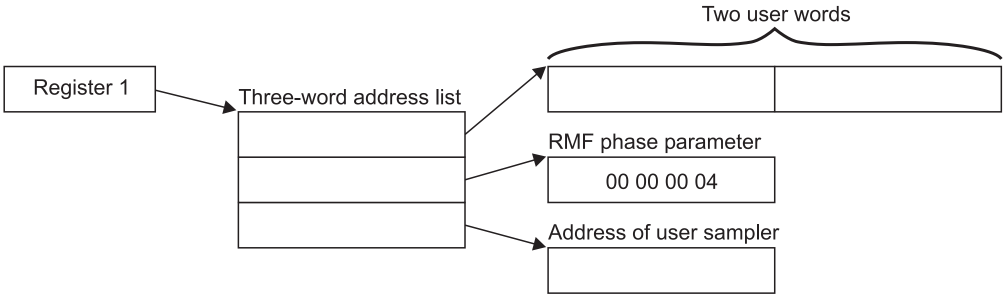


Figure 4-1. ERBMFIUC Input Parameter Structure

Sampling data at each cycle

To sample data at each cycle, the steps described earlier for initialization must be performed to load and page fix the user sampler routine. A user sampler routine is activated at each cycle only when another measurement that includes a sampling routine is activated. These measurements include paging activity, page/swap data set activity, channel path activity, I/O queuing activity, device activity, and trace activity. At least one of these measurements must be specified to enable RMF to invoke your user sampler.

When the sampler gets control, register 1 points to a two-word area. One of these words, selected by your installation, contains the address of the storage area obtained for the sampler by ERBMFIUC. Figure 4-2 illustrates the input parameter structure.

When the user sampler is entered, the system is in supervisor state, and all interrupts are disabled. The routine runs in key 0. It can sample any fixed data in CSA, SQA, or the nucleus; no other data areas can be sampled. You place the data sampled in the storage area obtained by ERBMFIUC and passed to you when your routine is invoked. This storage area is always in SQA (subpool 245). At the end of the RMF interval, RMF passes the address of the storage area to the user interval processing routine. Should your routine cause a page fault, the Monitor I session terminates abnormally with an abend code of 0FE.

When your sampling is completed, return control by branching on register 14.

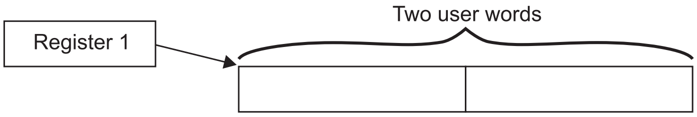


Figure 4-2. User Sampler Input Parameter Structure

Note: The user sampler must reside in SYS1.SERBLPA. See "Adding Your Routines to RMF" later in this chapter.

Interval processing

The interval processing user exit is ERBMFDUC. It is invoked at the start of the Monitor I session and at the end of each RMF interval.

When the exit gets control, register 1 points to a two-word address list. The first address points to the two-word area reserved for use by your routines. When these routines include a user sampler, one of these words, selected by your installation, will contain the address of the sampled data. The second address points to the RMF phase parameter. This parameter is a full word that contains X‘4’ when the exit is called during Monitor I session initialization, X‘8’ when the exit is called at the end of an RMF interval, or X‘C’ when the exit is called at the end of an RMF interval for which data collection was skipped. Figure 4-3 illustrates the input parameter structure.

When the interval processing exit routine is entered, the system is in supervisor state, and all interrupts are enabled. The routine runs in key 0. The routine can process the data generated by the user sampler. It can also collect its own data from system control blocks or system state indicators and format an SMF record. The SMF record can be written to the SMF data set; see *z/OS MVS System Management Facilities (SMF)* for details on using the SMFEWTM macro instruction to write a user SMF record.

The SMF record or a record your routine formats as agreed by convention between ERBMFDUC and ERBMFRUR (the report writer exit routine) can be printed by your report writer. Your routine can format SMF record output, report record output, or both. When your routine formats any records to be printed by your report writer, the address of the formatted records must be placed in the user word selected by your installation. Because the user words are passed to your report writer, the records can then be printed in a formatted report.

When the length of the RMF interval exceeds 99 minutes, which can occur when RMF is not dispatched at the end of an interval, data collection for the interval is skipped. Because there is no data collected, RMF does not call the report writer user exit (ERBMFRUR); instead, ERBMFDUC is called twice. The phase parameter is X‘8’ for the first invocation of the exit routine and X‘C’ for the second. When the exit routine is called with a phase parameter of X‘C’, your routine must free the storage areas normally freed by ERBMFRUR. RMF issues a message to notify the operator that data collection was skipped for the interval.

When your routine has completed processing, return control by branching on register 14.

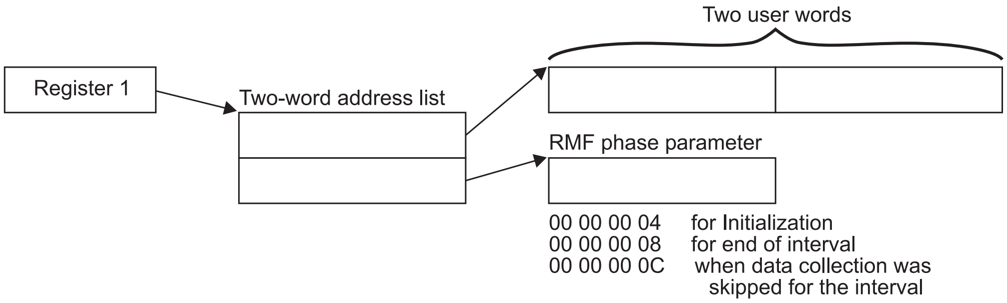

Two user words
Register 1
Two-word address list
RMF phase parameter
00 00 00 04 for Initialization
00 00 00 08 for end of interval
00 00 00 0C when data collection was skipped for the interval

Figure 4-3. ERBMFDUC Input Parameter Structure

Report writing during session processing

The report writer exit is ERBMFRUR. It is called once during the Monitor I session report writing phase.

Mon I reports

When the exit gets control, register 1 points to a two-word address list. The first address points to the two-word area reserved for use by user routines. The second address points to the RMF phase parameter, which is always X'10' for the report writer. Figure 4-4 illustrates the input parameter structure.

When the report writer exit is entered, the system is in problem state, and all interrupts are enabled. The routine runs in the user key 8. The user word your installation selects contains the address of the formatted records built by ERBMFDUC. Because all of your installation's exit routines use these words, the report writer must not alter their contents. Report writer processing must obtain output space for the printed reports, then write the reports for subsequent printing. Before terminating, the routine must free the storage that contained the records formatted by ERBMFDUC.

When the report writer completes its processing, return control by branching on register 14.

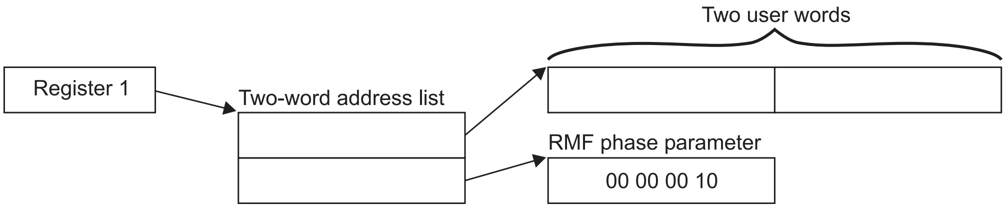


Figure 4-4. ERBMFRUR Input Parameter Structure

Termination

The termination exit is ERBMFTUR. It is called when the Monitor I session is terminated.

When the exit gets control, register 1 points to a two-word address list. The first address points to a two-word area reserved for use by your routines. The second address points to the RMF phase parameter, which is always X'C' for termination. Figure 4-5 illustrates the input parameter structure.

When the termination routine is entered, the system is in supervisor state, and all interrupts are enabled. The routine runs in key 0. You would use this exit to page free any user samplers or data areas and to free any user SQA data areas obtained by the other exits, with one exception: during termination processing, ERBMFTUR gets control before the report writer exit (ERBMFRUR). Therefore, it must free only the SQA and global storage the other user routines obtained, but it **must** not free the storage the interval processing routine (ERBMFDUC) used to build records to be passed to the report writer. The address of this storage will be in the user word selected by your installation.

When the termination routine has completed processing, return control by branching on register 14.

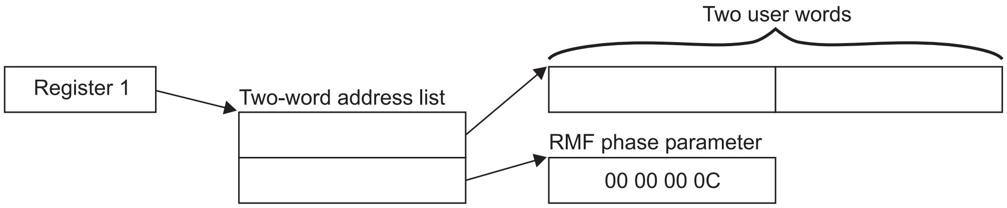


Figure 4-5. ERBMFTUR Input Parameter Structure

Tracing your own field

Whenever the TRACE option is in effect during a Monitor I session, you can use the tracing routines to trace the contents of any SQA, CSA, or nucleus field that you require. The EXITS option, required to generate your own reports, is not required to use the trace facilities because the trace exit, ERBTRACE, is called whenever the TRACE option is in effect.

The field that you want to trace must be fixed in CSA, SQA, or the nucleus; it cannot contain negative values; and it must be from one to four bytes in length. Once you have selected your field, there are two steps required to enable RMF to trace the contents of the field. After you have performed these steps, you can then specify the name in the field name portion of the TRACE option. The steps can be performed in any order, but both must be done before you can use RMF to trace the field.

Step 1 -- Defining the name to RMF

To define the name, you must add four fields to the RMF CSECT ERBMFTTB, which contains the names RMF recognizes as valid for tracing. The fields you must add are:

1. The name of the field to be traced. The name can be from one to eight bytes long. It must not be the same as any name already recognized by RMF. When the name of the field is less than eight bytes long, it must be padded on the right with blanks to a length of eight bytes.
2. The length of the name. This field is one byte long; the value must be from 01 to 08.
3. A one-byte constant that always contains the value X‘DC’.
4. The length of the field to be traced. This field is one byte long; the value must be from 01 to 04.

One byte of binary zeroes must follow the last entry to be added; the byte of binary zeroes indicates the end of the variable-length trace table. Figure 4-6 shows an example of how to superzap ERBMFTTB to add a new name for tracing. The parenthesized numbers in the text refer to the parenthesized numbers in the figure. The example adds a nucleus field named MYDATA (1) that is two bytes in length (4) to the list of names valid for RMF tracing. The name is six bytes long (2), and the required constant is also supplied (3). A byte of binary zeroes (5) indicates the end of the trace table. Adding the name definition to ERBMFTTB causes RMF to pass the name to ERBTRACE during each tracing cycle. The four fields must be added for each name you want RMF to trace; only the last entry must be followed by the byte of binary zeroes.

```
//ZAP        JOB          MSGLEVEL=1
//SS         EXEC         PGM=AMASPZAP
//SYSPRINT   DD           SYSOUT=A
//SYSLIB     DD           DSN=SYS1.SERBLINK,DISP=SHR
//SYSIN      DD           *
    NAME     ERBMFMFC         ERBMFTTB
    VER      040C             0040D7C1
    REP      040C             D4E8C4C1E3C14040 (1)
    REP      0414         06  (2)
    REP      0415         DC  (3)
    REP      0416         02  (4)
    REP      0417         00  (5)
/*
```

Figure 4-6. Example of Adding a Name to ERBMFTTB

Step 2 -- Replacing ERBTRACE

The tracing user exit is ERBTRACE. The function of ERBTRACE is to return to RMF the address of a valid user field. It is called by the RMF tracing routine whenever it encounters a trace name that is not the name of a field in the SRM domain table. To trace your own field, you must replace ERBTRACE with your own routine and link edit your ERBTRACE with the RMF CSECT ERBMFITR.

When ERBTRACE gets control, register 1 points to a two-word address list. The first address points to an eight-byte field that holds the name to be validated. The second address points to a full word to be used by ERBTRACE to return the address of the user field to RMF. Figure 4-7 illustrates the input parameter structure.

When ERBTRACE is entered, the system is in supervisor state, and all interrupts are enabled. The routine runs in key 0. It must examine the field name passed to it by RMF to determine if the name is a user field name. When the name is a valid user name, place the address of the field to be traced in the first parameter, set a return code of zero in register 15, and return control. If the name is not one recognized as a valid user name, always set a non-zero return code in register 15 before returning control. The non-zero return code tells RMF to process the name.

When your processing is completed, return control by branching on register 14.

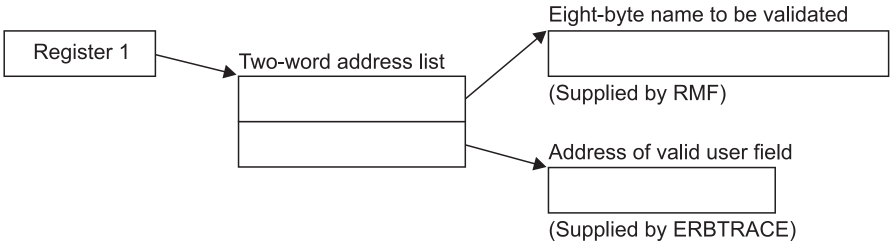


Figure 4-7. ERBTRACE Input Parameter Structure

Report writing by the Postprocessor

The Postprocessor user exit is ERBMFPUS. It is called during post-processing at the point when the Postprocessor first encounters each SMF record. When the exit gets control, register 1 points to a three-word address list:

- The first address points to a full word that contains the address of the SMF record to be processed.

- The second address points to a full word reserved for the use of your routine. The user word contains zeros the first time the exit is called, and the Postprocessor does not modify its contents between invocations of the user exit routine. Thus, the word can be used to save information, such as the address of a DCB, that is needed by a subsequent invocation of the user exit routine.
- The third address points to a FIXED(8) field, which contains X’01’ for EOF.

Figure 4-8 illustrates the input parameter structure.

When the Postprocessor user exit is entered, the system is in problem state and all interrupts are enabled. The routine runs in the user key 8.

Your routine examines the SMF record passed to you, performs any required processing, and set a return code in register 15. The return code depends on the action you want the Postprocessor to take. A return code of 0 tells the Postprocessor to continue processing the SMF record. A return code of 4 tells the Postprocessor to ignore the SMF record; set a return code of 4 when the exit routine has, for example, processed the record or determined that it should not be processed. A return code of 8 indicates that the Postprocessor should terminate.

The processing your exit performs can consist of formatting the data in the records that the interval processing user exit routine (ERBMFDUC) generates into a printed report. Your exit could also screen the SMF records that the Postprocessor encounters to determine which records are to be included in any reports generated by the Postprocessor, or it could use the SMF records RMF generates as input to your own report. Because all SMF records are passed to the user exit, ERBMFPUS could also be used to incorporate any SMF data reduction routines used at your installation into the RMF Postprocessor function.

When your routine has finished processing, set the appropriate return code in register 15 and return control to the RMF Postprocessor by branching on register 14.

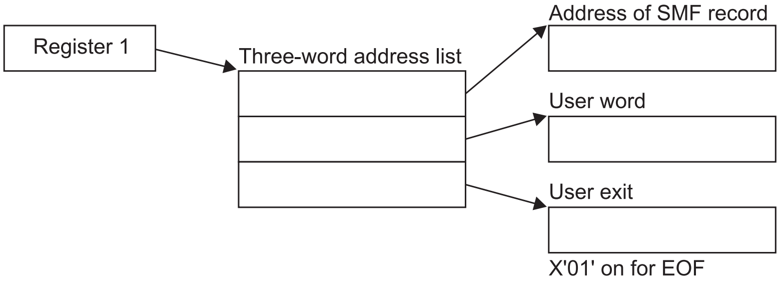


Figure 4-8. ERBMFPUS Input Parameter Structure

Adding your routines to RMF

Before your Monitor I session user exit routines can be tested and used, they must be assembled and link edited with the appropriate RMF modules. If you are using your private libraries, you have to ensure that they are concatenated in front of the distributed RMF libraries. Figure 4-9 shows sample JCL for performing the required link edit for all user routines except the sampler routine. If you have a user sampler, a separate link edit is required; a sample is shown in Figure 4-10.

Mon I linking

```
//LINKEXIT JOB   MSGLEVEL=1
//LINK0001 EXEC PGM=IEWL,PARM='MAP,XREF,REUS,RENT,REFR,NCAL'
//SYSPRINT DD    SYSOUT=A
//SYSLMOD  DD    DSN=SYS1.SERBLINK,DISP=(OLD,KEEP)
//SYSUT1   DD    UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD    *
    (ERBMFIUC object deck)
    ENTRY   ERBMFIUC
    NAME    ERBMFIUC(R)
    (ERBMFDUC object deck)
    ENTRY   ERBMFDUC
    NAME    ERBMFDUC(R)
    (ERBMFRUR object deck)
    ENTRY   ERBMFRUR
    NAME    ERBMFRUR(R)
    (ERBMFTUR object deck)
    ENTRY   ERBMFTUR
    NAME    ERBMFTUR(R)
    (ERBTRACE object deck)
    INCLUDE SYSLMOD(ERBMFITR)
    ENTRY   ERBMFITR
    NAME    ERBMFITR(R)
    (ERBMFPUS object deck)
    ENTRY   ERBMFPUS
    NAME    ERBMFPUS(R)
/*
```

Figure 4-9. Replacing Installation Exits

```
//LINKEXIT JOB   MSGLEVEL=1
//LINK0001 EXEC PGM=IEWL,PARM='MAP,XREF,REUS,RENT,REFR,NCAL'
//SYSPRINT DD    SYSOUT=A
//SYSLMOD  DD    DSN=SYS1.SERBLPA,DISP=(OLD,KEEP)
//SYSUT1   DD    UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN         DD    *
    (user sampler object deck)
    ENTRY   entry name
    NAME    sampler name
/*
```

Figure 4-10. Adding a User Sampler

End of Programming Interface information

Monitor II session user reports

Programming Interface information

RMF generates a Monitor II session report by invoking a data-gathering module and a data-reporting module in response to either:

- a menu item identifying a display session report
- an option identifying a background session report

From an external viewpoint, the menu item and the option are different because they are used during different types of sessions, have slightly different syntax, and produce either display output or printed output. However, from an internal point of view, the menu item and the option are very similar. The valid menu items for a display session are listed in the RMF CSECT ERBFMENU.

Note: If you are running the Kanji version of RMF, the corresponding CSECT is ERBJMENU, and you should ensure that both CSECTs stay synchronized.

The options for a background session are listed in the RMF CSECT ERBBMENU. The formats of the entries in each list are identical. When an option or menu item is specified during a session, RMF uses the data entry for the report in the list appropriate for the session type to verify that the option or menu item is valid and to load the required data gatherer and data reporter modules.

Each list contains an entry called USER that enables you to add a single user report. When USER is specified, RMF loads modules ERBGUS99, the data gatherer for USER, and ERBRUS99, the data reporter for USER. By replacing these two modules with your own routines, you can add a single report to the Monitor II reports provided by RMF. This process is described later in this chapter under “Coding a User Report.”

The data gathering module and the data reporting module communicate through a type 79 SMF record. The data gatherer formats the record and completes the required data fields. The data reporter uses the data in the record to generate a formatted report for printing or display. See “SMF Record Type 79.”

To add more than one Monitor II session report, you must, in addition to providing a data gatherer and a data reporter, add an entry to ERBFMENU for a display session report and to ERBBMENU for a background session report. Then, when your option or menu item is specified during a session, RMF will load your data gatherer and data reporter to generate the report. The process to follow to add an entry to the option list and menu list is described later in this chapter under “Installing a User Report.”

Guidelines

Each of the user functions is described in detail in the following sections. The following guidelines apply to all Monitor II user exit routines.

- All of the user exit routines must be reenterable.
- All user-written exit routines receive control in 31-bit addressing mode.
- The routines must save registers when they receive control and restore registers when they return control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.
- All of the user exit routines receive control in problem state, key 8.

SMF record type 79

SMF record type 79 must be used to record data gathered by a user data gathering routine. Figure 4-11 shows the layout of the record sections that are common to all Monitor II data gatherers, whether coded by a user or provided by RMF. The figure illustrates the layout of these common sections by showing the expansion of the RMF mapping macro ERBSMF79.

The fields in the common sections fall into three categories. Each category is indicated by a letter in the figure that corresponds to the letters in the following text:

SMF record

A The fields that the RMF routines fill in before the data gathering routine is invoked.

B The fields that the data gathering routine must fill in during its processing. (See “Relocate Blocks” later in this section.)

C The fields that the RMF routines will fill in when the RECORD option is in effect. RMF completes these fields after the data gatherer returns control but before the record is written to the SMF data set. During a display session or a background session when NORECORD is in effect, these fields are not completed because the record is not actually written to the SMF data set.

Before invoking the data gatherer, RMF calculates the length of the storage buffer required for the record, as described later under “Relocate Blocks,” obtains a buffer for the record, and fills in some of the common section fields. The address of the SMF record buffer is passed to the data gatherer. The data gatherer fills in some fields in the common section and all of the data section of the record.

```
    ********************  COMMON SMF HEADER  *****************************
        SMF79HDR   DSECT
      C SMF79LEN   DS      BL2      RECORD LENGTH
        SMF79SEG   DS      BL2      SEGMENT DESCRIPTOR
      C SMF79FLG   DS      BL1      HEADER FLAG BYTE
        SMF79RRF   EQU     X'80'    NEW SMF RECORD FORMAT IF=1
        SMF79SUT   EQU     X'40'    SUBTYPE UTILIZED IF=1
        SMF79ESA   EQU     X'08'    MVS/ESA IF=1
        SMF79VXA   EQU     X'04'    MVS/XA IF=1
        SMF79OS    EQU     X'02'    OPERATING SYSTEM IS OS/VS2
        SMF79BFY   EQU     X'01'    SYSTEM IS RUNNING IN PR/SM MODE
        SMF79PTN   DS      BL1      PR/SM PARTITION NUMBER
      C SMF79RTY   DS      BL1      RECORD TYPE
        SMF79TME   DS      BL4      TOD RECORD WRITTEN
        SMF79DTE   DS      PL4      DATE RECORD WRITTEN
      C SMF79SID   DS      CL4      SYSTEM ID FROM INSTALLATION
      C SMF79SSI   DS      CL4      SUBSYSTEM ID (RMF)
      B SMF79STY   DS      BL2      SUBTYPE
      A SMF79TRN   DS      BL2      NUMBER OF TRIPLETS IN THIS RECORD
                   DS      BL2      RESERVED
      A SMF79PRS   DS      BL4      OFFSET TO RMF PRODUCT SECTION
      A SMF79PRL   DS      BL2      LENGTH OF RMF PRODUCT SECTION
      A SMF79PRN   DS      BL2      NUMBER OF RMF PRODUCT SECTIONS
      **********************  INDIVIDUAL HEADER EXTENSION *******************
      A SMF79MCS   DS      F -      OFFSET TO MONITOR II CONTROL SECTION
      A SMF79MCL   DS      H -      LENGTH OF MONITOR II CONTROL SECTION
      A SMF79MCN   DS      H -      NUMBER OF MONITOR II CONTROL SECTION
      B SMF79ASS   DS      F -      OFFSET TO DATA SECTION
      B SMF79ASL   DS      H -      LENGTH OF DATA SECTION
    B A SMF79ASN   DS      H -      NUMBER OF DATA SECTION
      A SMF79DCS   DS      F -      OFFSET TO DATA CONTROL SECTION
      B SMF79DCL   DS      H -      LENGTH OF DATA CONTROL SECTION
      B SMF79DCN   DS      H -      NUMBER OF DATA CONTROL SECTION
        SMF79QSS   DS      F -      OFFSET IOQ GLOBAL SECTION
        SMF79QSL   DS      H -      LENGTH IOQ GLOBAL SECTION
        SMF79QSN   DS      H -      NUMBER IOQ GLOBAL SECTION
```

Figure 4-11. ERBSMF79 Mapping Macro Expansion (Part 1 of 2)

```
 *********************  COMMON SMF PRODUCT SECTION ********************
  SMF79PRO   DSECT
C SMF79MFV   DS      CL2      RMF VERSION NUMBER, WITH
  *                                    INTRODUCTION OF THE MVS
  *                                    SOFTWARE LEVEL, THE FORMAT
  *                                    CHANGES TO PACKED (VRLF),
C SMF79PRD   DS      CL8      PRODUCT NAME
  SMF79IST   DS      PL4      TOD MONITOR 1 INTERVAL START: OHHMMSSF
C SMF79DAT   DS      PL4      DATE MONITOR 1 INTERVAL START: 00YYDDDF
  SMF79INT   DS      PL4      DURATION OF MONITOR 1 INTERVAL: MMSSTTTF
  *          DS      BL2      RESERVED
B SMF79SAM   DS      BL4      NUMBER OF SAMPLES
  *          DS      BL2      RESERVED
  SMF79FLA   DS      BL2      FLAGS
  SMF79ISS   DS      X'40'    INVALID SAMPLES TO BE SKIPPED
  SMF79M3R   DS      X'20'    RECORD WAS WRITTEN BY RMF MONITOR III
  SMF79ISM   DS      X'10'    INTERVAL WAS UNDER SMF CONTROL
  *          DS      BL4      RESERVED
B SMF79CYC   DS      PL4      CYCLE IN PACKED DECIMAL 000TTTTF
B SMF79MVS   DS      CL8      MVS SOFTWARE LEVEL
B SMF79IML   DS      BL1      TYPE OF PROCESSOR COMPLEX ON WHICH DATA IS MEASURED
B SMF79PRF   DS      XL1      PROCESSOR FLAGS
B SMF79QES   EQU     X'80'    EQUIPPED WITH EXPANDED STORAGE
B SMF79CNE   EQU     X'40'    EQUIPPED WITH ESCON CHANNEL
B SMF79DRC   EQU     X'20'    ESCON DIRECTOR IN CONFIG.
B SMF79EME   EQU     X'10'    SYSTEM IS RUNNING IN Z/ARCHITECTURE
B SMF79PTN   DS      BLI      PR/SM PARTITION NUMBER
  SMF79SLR   DS      BL1      SMF RECORD LEVEL
  SMF79IET   DS      CL8      INTERVAL EXPIRATION TIME TOKEN
 *********************  MONITOR II CONTROL SECTION  *******************
  R79CHL     DSECT            COMMON RECORD 79 HEADER
B R79GTOD    DS      XL4 -    DATA GATHERER CALL TOD
B R79LF2     DS      XL1 -    FLAG BYTE
  R79PAR     EQU     X'80'    NOT ENOUGH RELOCATE SECTION TO
  *                           COMPLETE DATA GATHERING
  R79SG      EQU     X'40'    REPORT TO BE SORTED BY SG
  R79RV1     DS      XL1 -    RESERVED
C R79SES     DS      CL2 -    SESSION NAME
  R79RSV     DS      XL2 -    RESERVED
  R79USER    DS      XL2 -    USER FIELD
C R79RID     DS      CL8 -    MEASUREMENT NAME
C R79CTXTL   DS      XL2 -    LEN OF COMMAND TEXT
C R79CTEXT   DS      CL32 -   COMMAND TEXT
C R79DTXTL   DS      XL2 -    LEN OF DEFAULT DR TEXT
C R79DTEXT   DS      CL32 -   DEFAULT DR TEXT
C R79IST     DS      CL4 -    MON III INTERVAL START TIME :0HH MMSSF
  ********************* DATA  SECTION  ********************************
  R799LCU    DS      BL2      LOGICAL CONTROL UNIT NUMBER 0 TO 255
  R799SGN    DS      CL8      STORAGE GROUP NAME
```

Figure 4-11. ERBSMF79 Mapping Macro Expansion (Part 2 of 2)

Relocate blocks

The data section of SMF record type 79 is unique to each report. It is composed of one or more data sections called **relocate** blocks and, possibly, one data control section. A relocate block is the portion of the SMF record that contains the data for one report data line. A record for a row report has one relocate block. A record for a table report has multiple relocate blocks; for example, the SMF record for the address space state data report includes one relocate block for each address space included in the report. When your SMF record has multiple relocate blocks and you are gathering data that applies to all of them, you can, instead of reporting the data in each relocate block, place this common data in a data control section, as described later under “Data Control Section”.

SMF record

The format of the data in the relocate block depends on the report you are generating. You set the format that best meets your needs. When you are generating a table report, the SMF record consists of multiple relocate blocks, and each relocate block must have the same length.

When you add a menu item to ERBFMENU or an option to ERBBMENU, the entry that describes the new report must include a field that specifies the length of the relocate block, the maximum number of possible relocate blocks, and the length of the data control section. For information on how to add an entry to ERBFMENU or ERBBMENU, see "Using the PICTURE macro" on page 4-24. To determine the storage to allocate, RMF multiplies the length of the relocate block by the maximum number of relocate blocks and adds this value to the length of the data control section and the common section. The result of this computation is the maximum possible length of the SMF record, and RMF allocates a buffer for the record that is equal in size to the maximum length.

To determine the actual length of the SMF record, the data gatherer must complete the fields in the individual header extension section that describe the offset, length, and number of data sections and the data control sections. After the data gatherer has completed its processing and returned control, RMF uses these values to determine the length of the SMF record to be written to the SMF data set, a calculation that is performed only when the RECORD option is in effect for a background session. Note that the value your routine sets in SMF79ASL and the value specified for RBLEN in the PICTURE macro for the report should be identical.

Other fields in the common section that the data gather completes are R79GTOD and SMF79STY. R79GTOD must contain a packed decimal value that indicates the time when the data gatherer was invoked, in the form 0hhmmssF, where F is the sign. SMF79STY can contain the subtype number of the SMF record that you are creating. You use this number as a unique identifier for each record subtype that you create; no subtype number should be less than 1000.

The maximum length of an SMF record is 32,756 bytes; any records that exceed this length are truncated before they are written to the SMF data set. Truncation, which can occur only during a background session when the RECORD option is in effect, occurs at the last relocate block boundary within the maximum length. When truncation occurs, RMF adjusts the field indicating the capacity of the buffer (SMF79ASN) to indicate the actual number of relocate blocks in the record. If no truncation occurs, RMF leaves SMF79ASN unchanged.

Data control section

A data control section is useful when your SMF record might have many relocate blocks and some of the data you are gathering is common to all of them. For example, the channel path Monitor II control section (subtype C) uses a control section to record the number of times the channel was sampled. To use a data control section:

1. Set the value for the FBLEN parameters on the PICTURE macro instruction for your report, as described under "Using the PICTURE macro" on page 4-24.
2. Format the data control section to hold the common data.
3. Place it between the Monitor II control section and the data section. SMF79DCS contains the offset at which it should start.
4. Set SMF79DCL and SMF79DCN to the length and number of the data control sections.

5. Set the offset to the first data section SMF79ASS to point to the end of the data control section.

When a data control section is *not* used:

1. Set SMF79ASS to the value in SMF79DCS.
2. Set SMF79DCL and SMF79DCN to 0.

Coding a user report

To add a Monitor II report, you must code your own data gatherer module and data reporter module. These modules can reside in SYS1.SERBLINK, SYS1.SERBLPA, a steplib, a joblib, a tasklib, or a library in a linklist.

The primary means of communicating data between the gatherer and the reporter is the type 79 SMF record. The gatherer collects data from whatever areas it can access (it runs in problem state with a key of 8) and places the data in the SMF record. The reporter takes the data from the SMF record, formats it for output, and passes it to the RMF putline routine. During a Monitor II background session, the data reporter would be called when the REPORT option is in effect. When NOREPORT and RECORD are in effect, RMF writes out the SMF records that the data gatherer formats, and the data reporter is not invoked. Your data reporter can be invoked at a later time by the Postprocessor.

A Monitor II session report can have operands that the report user specifies when requesting the report. Any operands specified when a report is requested are passed to both the data gatherer and the data reporter. The defaults established for each possible operand are specified in the option list or menu list entry for the report; these defaults are also passed to both the data gatherer and the data reporter. Your routines can also include hard-coded default operands.

Because the option list and menu item list are in different RMF control sections, you can set different default operands for a background session and a display session. Each list entry contains separate fields for the data gatherer default operands and the data reporter default operands; you can thus set different default operands for the data gatherer and the data reporter. For example, the default operands for the RMF address space state data gatherer module cause data to be gathered on all address spaces in the system; to limit the actual output produced, the defaults for the reporter cause only the active address spaces to be reported. "Using the PICTURE Macro" describes how to specify default operands.

RMF passes parameters to both the gatherer and reporter; these parameters include a subpool number that indicates the subpool from which the routines should obtain the storage they require, and two user words that can be used for communication between the data gatherer and the data reporter. Because the same two words are passed to both routines, use of these words must be governed by conventions established by your installation.

Note: A system status line precedes each display report supplied by IBM. RMF obtains the data for this line before it invokes the data gatherer for the report. RMF will generate the same system status line before each user-coded display report.

Data gatherer

The data gatherer runs in problem state, with a key of 8, and in 31-bit addressing mode. The data gatherer must be reenterable. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.

Upon entry to the data gatherer, register 1 points to a contiguous list of seven addresses that point to seven input parameters. The first address points to the first parameter, the second address points to the second parameter, and so forth. The input parameters are:

First Parameter: A fullword entry code that must always be X'2'.

Second Parameter: The operands, if any, specified by the report user when he requested the report, in the form:

LL	text

LL A two-byte length field indicating the length of the following text (does not include the two bytes of LL)

text A character string of up to 32 characters containing the input operands

When the report has no operands or the report request did not include operands, LL is set to zeros.

RMF determines the operands to be placed in **text** by scanning the report request. The first non-blank character after the report name is assumed to be the first character of the operand field. The next blank character is assumed to mark the end of the operand field.

Third Parameter: The default operands from ERBFMENU or ERBBMENU, in the form:

LL	text

LL A two-byte length field indicating the length of the following text (does not include the two bytes of LL)

text A character string of up to 32 characters containing the default operands

When the report has no operands or no default operands, LL is set to zeros.

Fourth Parameter: The pointer to the SMF record buffer where your routine is to place the data it gathers.

Fifth Parameter: The first of the two words reserved for the use of your routines.

Sixth Parameter: The second of the two words reserved for the use of your routines.

Seventh Parameter: A byte containing the number of the subpool to use when you issue a GETMAIN to obtain the storage your routine requires.

The processing your data gathering routine performs is determined largely by the nature of the report for which you are gathering data. This processing should include a validation of the entry code in the first parameter to verify that it is X'2'. If it is not, set a return code of 8 in register 15 and return control.

If the report has operands that can be specified when the report is requested, check the second input parameter to determine if the request specified operands. If it did, validate the syntax of the operands; if the syntax is invalid, set a return code of 4 in register 15 and return control. If the request did not specify operands, verify the syntax of the default menu operands passed as the third input parameter; if the syntax is invalid, set a return code of 24 in register 15 and return control.

Your routine should complete the required fields in the SMF record common section (the B fields in Figure 4-11), using the RMF mapping macro ERBSMF79 to access the fields in the common section. The address of the storage buffer obtained for your record is passed in the fourth input parameter. Your routine would gather the data required and format the data section of the record as agreed upon by convention between the data gatherer and the data reporter. Should your routine locate no data that is applicable to the report requested, set a return code of 16 in register 15 and return control.

When your routine has finished processing, set a return code in register 15 and return to the caller by branching on the contents of register 14. Table 4-1 shows the possible return codes, their meaning, and the action RMF takes in response. These return codes apply to both the data gatherer and the data reporter.

Note: If your report will be run only during a display session, you can perform both the data gathering function and the data reporting function in the data reporter module. In this case, your data gatherer's only function would be to set a return code of zero in register 15. However, if you choose to perform both functions in the data reporter module, your report cannot run during a Monitor II background session and, during a display session, you will not be able to use the recall command to re-display your report.

Table 4-1. Return Codes from the Data Gatherer and Data Reporter

Code	Meaning	RMF Response (Display Session)	RMF Response (Background Session)
0	Successful completion.	The session continues.	The session continues.
4	Invalid operand syntax.	The command is displayed as entered.	Message ERB409I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues. The operator can modify the session options.
8	Invalid entry code.	Abend - the user code is 1402.	Abend - the user code is 1402.
12	I/O error.	Messages ERB403I and ERB404I are displayed, including the SYNAD text.	The current measurement continues when RECORD is in effect, but no subsequent reports are printed; otherwise, the measurement is discontinued. The session continues.
16	No data found.	Message ERB405I is displayed.	Message ERB405I is issued. No report or SMF record is produced for this interval. All measurements continue.

Table 4-1. Return Codes from the Data Gatherer and Data Reporter (continued)

Code	Meaning	RMF Response (Display Session)	RMF Response (Background Session)
20	ESTAE macro failed.	Message ERB406I is displayed.	Message ERB406I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues.
24	Menu default operand syntax error.	Message ERB407I is displayed, including the menu defaults and advice to retry the report, specifying all operands.	Message ERB407I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues.
28	The amount of data to be gathered exceeds the number of available relocate blocks.	Message ERB411I is displayed.	Message ERB411I is issued. The report or SMF record produced for the interval includes only the data gathered before the condition was detected. All measurements continue.
32	Monitor I report not active.	Message ERB412I is displayed.	Message ERB412I is issued. No report or SMF record is produced for the interval. All measurements continue.
36	Monitor I interval is less than Monitor II interval.	Message ERB413I is displayed.	Message ERB413I is issued. No report or SMF record is produced or the interval. All measurements continue.
40	The SRM’s store channel path status facility is not active. Used by channel path activity (CHANNEL) report.	Message ERB264I is displayed.	Message ERB264I is issued. No report or SMF record for channel path activity is produced; the current measurement is discontinued. All other measurements continue.
44	Report option not applicable in goal mode.	Message ERB434I is displayed.	Message ERB434I is issued. No SMF record is produced for this report. All other measurements continue.
48	No transaction data available.	Message ERB435I is displayed.	Message ERB435I is issued. No SMF record is produced for this report. All other measurements continue.
52	SRM mode changed - interval skipped.	Message ERB436I is displayed.	Message ERB436I is issued. No SMF record is produced for this report. All other measurements continue.
56	Report option not applicable in compatibility mode.	Message ERB434I is displayed.	Message ERB434I is issued. No SMF record is produced for this report. All other measurements continue.
>56	Unexpected.	Message ERB408I is displayed.	Message ERB408I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues.

Data reporter

The data reporter runs in problem state, with a key of 8, and in 31-bit addressing mode. The data reporter must be reenterable. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.

The data reporter formats each line in the report, using the data placed in the type 79 SMF record by the data gatherer. The RMF putline routine is used to perform the actual output operation.

Because the putline routine handles the actual output operations, your data reporter can function identically during a background session, a display session, a display session in hardcopy mode, or an execution of the Postprocessor. The putline routine writes the line to a logical screen buffer for a display session, to a logical screen buffer and an output data set for a display session in hardcopy mode, or to an output data set for a background session or an execution of the past processor. For a display session, the screen is updated to show the lines collected by the putline routine when your data reporter returns control. Note that RMF handles any framing required for the display session user to view all the frames in a multi-frame table report after the data reporter completes its processing.

The data reporter you code can generate either a row report or a table report. The maximum number of header lines is two.

A row report consists of one or two header lines and a single data line. For a row report, RMF invokes the data reporter twice: once to format the header line(s) and once to format the data line. When a row report is executed repetitively, RMF invokes the reporter to format the header line(s) for the first execution; for all subsequent executions, the reporter is invoked to format a data line.

A table report consists of one or two header lines and a variable number of data lines. For a table report, RMF invokes the data reporter once to format both the header line(s) and the data lines. The number of data lines must be less than or equal to the number of relocate blocks created in the SMF record by the data gatherer.

Upon entry to the data reporter, register 1 points to a contiguous list of eleven addresses that point to eleven input parameters. The first address points to the first parameter, the second address points to the second parameter, and so forth. The input parameters are:

First Parameter: A full word entry code that can be either X‘1’ or X‘2’. X‘1’ indicates that the reporter is to format the header line(s) for a row report. X‘2’ indicates, for a row report, that the reporter is to format the single data line. For a table report, the entry code should always be X‘2’, indicating that the reporter is to format both the header line(s) and the data lines.

Second Parameter: A full word report mode indicator that can have either of the following values:

X‘1’ Total mode; the values in the report are to reflect session totals.

X‘2’ Delta mode; the values in the report are to reflect changes since the last request for the report.

Third Parameter: The operands, if any, specified by the report user when he requested the report, in the form:

LL	text

LL A two byte length field indicating the length of the following text (does not include the two bytes of LL).

text A character string of up to 32 characters containing the report operands.

When the report has no operands or the report request did not include operands, LL is set to zeros.

Fourth Parameter: The default operands from ERBFMENU or ERBBMENU, in the form:

LL	text

LL A two byte length field indicating the length of the following text (does not include the two bytes of LL).

text A character string of up to 32 characters containing the default operands.

When the report has no operands or no default operands, LL is set to zeros.

Fifth Parameter: The address of the current SMF record buffer; that is, the buffer where the data gatherer has placed the data for the current execution of the reporter.

Sixth Parameter: The address of the previous SMF record buffer; that is, the buffer where the data gatherer placed the data for the previous execution of the report. When the report mode (the second parameter) indicates delta mode, the data fields in the previous SMF record enable your data reporter to calculate the changes that have occurred since the last request for the report.

Seventh Parameter: The first of the two words reserved for the use of your routines.

Eighth Parameter: The second of the two words reserved for the use of your routines.

Ninth Parameter: A byte containing the number of the subpool to use when you issue a GETMAIN to obtain the storage your routine requires.

Tenth Parameter: The address of the RMF putline routine. When the data reporter has formatted a report line, it calls the putline routine to perform the actual output operation.

Eleventh Parameter: The control block address that your data reporter must pass to the putline routine.

The processing your data reporting routine performs is determined largely by the nature of the report for which you are formatting report lines. This processing should include a validation of the entry code. If it is not a valid code, set a return code of 8 in register 15 and return control. If your report is a row report, examining the entry code determines whether your routine has been invoked to format the header line(s) or the data line for the report.

If the report has operands that can be specified when the report is requested, check the third input parameter to determine if the request specified operands. If it did, validate the syntax of the operands; if the syntax is invalid, set a return code of 4 in register 15 and return control. If the request did not specify operands, verify the syntax of the menu default operands passed as the fourth input parameter; if the syntax is invalid, set a return code of 24 in register 15 and return control.

If your report contains fields that are affected by the session mode – either delta mode or total mode – check the second input parameter to determine which mode is in effect. When delta mode is in effect, use the data fields in the previous SMF record buffer (pointed to by the sixth parameter) and the data fields in the current SMF record buffer (pointed to by the fifth parameter) to calculate the changes that have occurred since the last report request.

When your routine has formatted a report line, it should invoke the RMF putline routine to perform the actual output operation. To use the putline routine, perform the following steps:

1. Set up the input parameters that the putline routine requires. To do this, set register 1 to point to a list of four addresses that point to the following four parameters:

   First Putline Parameter: The record you have formatted, preceded by a two-byte length field. The length specified **must not** include the two bytes of the length field. The maximum record length is 79 characters. Note that the 3270 field attribute bytes must **not** be included; RMF supplies these bytes.

   Second Putline Parameter: A two-byte field that tells the putline routine whether the record you have formatted is a header line or a data line. The field must contain one of the following:

   'HD' Indicates that the record is a header line

   'DT' Indicates that the record is a data line

   Header lines generally contain column headings. These lines are repeated when the terminal user frames forward through a multi-frame table report or when the hardcopy output crosses a page boundary.

   Third Putline Parameter: A one-byte field; its bits have the following meaning:

   Bit **Meaning**

   0 Set to 1 if high intensity display is desired. Set to 0 if low intensity display is desired. (The bit is ignored during a background session.)

   1-7 Reserved. These bits must be set to zeros.

   Fourth Putline Parameter: The control block address that RMF passed to your data reporter in the eleventh input parameter.
2. Invoke the putline routine using standard linkage conventions. Set register 13 to point to your register save area, set register 15 to the address of the putline routine (passed to your data reporter in the tenth parameter), and pass control to the putline routine by a BALR 14,15 instruction.
3. When the putline routine returns control to the data reporter, a return code is set in register 15. A return code of zero indicates successful completion. A return code of 4, indicates an uncorrectable I/O error; set a return code of 12 in register 15 and return control.

Mon II reporting

When your data reporter has finished processing, set a return code in register 15 and return control by branching on the contents of register 14. Table 4-1 shows the possible return codes, their meaning, and the action RMF takes in response to each code.

Installing a user report

Once your data gatherer and data reporter are coded, two steps are required to install the report:

1. Include an entry for the report in the option list for a background session (ERBBMENU) and the menu list for a display session (ERBFMENU), depending on the type of session during which your report can be run.

   If data collected during a Monitor II background session is to be reported during execution of the Postprocessor, a copy of the option list control section (ERBBMENU) that includes the entry for your report must be link edited with the Postprocessor.

   RMF supplies the PICTURE macro to simplify the process of adding or changing an entry in the option list or menu list. See “Using the PICTURE macro” on page 4-24. You can also superzap an entry to make changes when the length of the entry is not changed.
2. Link edit your data gatherer and data reporter and test your report.

The option list or menu list consists of a set of variable-length entries, each describing a valid report. The option list appears in the RMF control section ERBBMENU; the menu list appears in ERBFMENU. Two separate control sections are provided to allow for a report that will run only during a background session or only during a display session. Also, the two different control sections allow different sets of default operands to be established for display sessions and background sessions. For example, you might want the display defaults to specify a limited set of possible data, while the background defaults specify all possible data.

The steps required to add an entry to the list are:

1. Determine whether the USER entry supplied by RMF is appropriate for your report. The USER entry contains specifications for a table report (RPTTYP=T) with a single relocate block (MAXRBS=1) that is four bytes long (RBLEN=4). The report title is ‘USER PICTURE’. If the entry is not appropriate for your report, replace the entry with a new entry for USER.
2. If you are changing the USER entry or adding a new entry, make a copy of ERBFMENU for a display report or ERBBMENU for a background session — or both — from the source code data set.
3. In the copy you have made, either replace the USER entry or insert a new PICTURE macro. For a new display report, insert the PICTURE macro where you want the new report to appear in the menu frame. For details, see “Using the PICTURE macro” on page 4-24.
4. Assemble ERBFMENU for a display report and ERBBMENU for a background report.
5. Link edit the menu list or option list CSECT(s) that you have assembled into the RMF load modules:

   `ERBMFMFC` - RMF control
   `RMFMON` - Monitor II RMFMON command
   `ERBRMFPP` - Postprocessor
   `ERB2RCTL` - Monitor II ISPF version
   `ERB2XDG0` - Monitor II Internal Data Gatherer

A sample of the control statements required is:

```
//LINKEXIT JOB   MSGLEVEL=1
//LINK0001 EXEC PGM=IEWL,PARM='MAP,XREF,REUS,RENT,REFR,NCAL'
//SYSPRINT DD    SYSOUT=A
//SYSLMOD  DD    DSN=SYS1.SERBLINK,DISP=(OLD,KEEP)
//SYSUT1   DD    UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD    *
  (ERBFMENU object deck)
  (ERBBMENU object deck)
  INCLUDE SYSLMOD(ERBMFMFC)
  ENTRY ERBMFMFC
  ALIAS ERBMFMPR
  ALIAS ERBMFCLS
  SETCODE AC(1)
  NAME ERBMFMFC(R)
  (ERBFMENU object deck)
  INCLUDE  SYSLMOD(ERBMFTSO)
  ENTRY    ERBMFTSO
  ALIAS    RMFMON
  NAME     ERBMFTSO(R)
  (ERBBMENU object deck)
  INCLUDE  SYSLMOD(ERBRMFPX)
  ENTRY    ERBRMFPP
  NAME     ERBRMFPX(R)
  (ERBFMENU object deck)
  INCLUDE SYSLMOD(ERB2RCTL)
  ENTRY ERB2RCTL
  NAME  ERB2RCTL(R)
  (ERBFMENU object deck)
  INCLUDE SYSLMOD(ERB2XDG0)
  ENTRY ERB2XDG0
  NAME  ERB2XDG0(R)
/*
```

Figure 4-12. Install User Report

To install your report, you must link edit your data gatherer and data reporter.

If you are using the USER entry, name your gatherer routine ERBGUS99; name your reporter routine ERBRUS99. Replace the dummy RMF modules that have these names with your own routines. The link edit control statements required are:

```
(ERBGUS99 object deck)
ENTRY ERBGUS99
NAME ERBGUS99(R)
(ERBRUS99 object deck)
ENTRY ERBRUS99
NAME ERBRUS99(R)
```

If you are not using the USER entry, give your data gatherer and data reporter modules names that match the names you are specifying in the PICTURE macro for the report that you are adding. Link edit the modules as shown in the above control statements, replacing ERBGUS99 with the name of your data gatherer and ERBRUS99 with the name of your data reporter.

Once your modules have been link edited, you are ready to test your report. You might find it simpler to test your new report on TSO before making it available to other RMF users at your installation. Perform the following steps:

1. Use a testing tasklib, a special partitioned data set (for example, TESTLIB.LOAD). Place your data gatherer, data reporter, and the RMFMON load module that includes the new menu list in the testing tasklib.

2. You can then test the new report by entering:

   ```
   CALL TESTLIB(RMFMON)
   ```

   The new menu should appear on the screen in response to this command. You can then invoke your report by specifying its menu item name.

If your report routine terminates abnormally, you can obtain a dump by replying 'STOP' to the messages describing the abnormal termination.

Using the PICTURE macro

The PICTURE macro describes a Monitor II session report to RMF. Use the PICTURE macro to replace the USER description or add or replace any entry in either ERBBMENU or ERBFMENU. The PICTURE macro is located in SYS1.MACLIB.

The syntax of the macro and the meaning of each operand are as follows:

```
[label]   PICTURE
        ID=name,
        GATHER=gathername,
        REPORT=reportname,
        RBLEN=length,
        RPTTYP={R|T}
        [,PFK=n]
        [,TITLE='title']
        [,DGTEXT='dgdefaults']
        [,DRTEXT='drdefaults']
        [,MAXRBS=nn]
        [,FBLEN=len]
        [,HELP={'*'|'panelname'}]
        [,WLMMODE={BOTH|COMPAT|GOAL}]
```

Figure 4-13. Syntax of the PICTURE Macro

ID=name
The option or menu item that will identify the report.

The name must consist of one to eight alphameric characters. The first character must not be 'R'; RMF takes 'R' to be a request to recall a report. For a display report, this name will appear on the menu frame.

GATHER=gathername
The name of the module RMF is to invoke to gather data for the report.

PFK=n
The PF key number associated with the report, where n is a one-digit or two-digit decimal identifier in the range of 1 to 24. For a display report, this number appears in the menu frame. If a PF key is not specified, the report is not associated with a PF key.

REPORT=reportname
The name of the module RMF is to invoke to format the header lines and data line(s) for the report.

RBLEN=length
The length of the relocate block generated by the data gatherer for each line in the report.

RPTTYP={R|T}
The type of report. T indicates a table report; R indicates a row report.

TITLE='title'
An optional report title. The title specified appears in the menu frame for a display session. The title must be enclosed in single quotes. Use two quotes to represent any quote used in the title. The title can contain up to 50 printable characters. However, a maximum of 35 characters can be printed or displayed; therefore, a title longer than 35 characters will be truncated to fit into the menu frame.

DGTEXT='dgdefaults'
The default operands that are passed to the data-gathering routine for the report. This field is optional; it is used when the report requires operands. The text must be enclosed in single quotes, and the maximum length of the text is 32 characters. Any characters are valid between the quotes. Use two quotes to represent any quote used in the text. When more than 32 characters are specified, the text is truncated.

DRTEXT='drdefaults'
The default operands that are passed to the data-reporting routine for the report. This field is optional; it is used when the report requires operands. The text must be enclosed in single quotes. Use two single quotes to represent any quotes used in the text. Any characters are valid between the quotes. When more than 32 characters are specified, the text is truncated.

MAXRBS=nnn
The initial number of relocate blocks. This number is equivalent to the maximum number of data lines in the report. The field is optional; when it is omitted, the default is 1 when RPTTYP=R is specified, indicating a row report. When RPTTYP=T is specified, indicating a table report, the field defaults to zero; however, enough storage is provided to allow a relocate block for each address space possible in the system. The maximum value possible for MAXRBS is 32,767.

FBLEN=len
The total length of all data control sections of the SMF record. The default value is 0.

HELP={'*'|'*panelname*'}]
Name of ISPF panel (maximal 8 characters) that contains help for this report. If HELP is requested on this report during a Monitor II ISPF display session, the panel '*panelname*' will be shown, if there is no message pending. If this option is omitted, '*' is generated by default which causes the tutorial displayed in such a case. The option has no effect for the TSO RMFMON session and for background sessions.

WLMMODE={BOTH|COMPAT|GOAL}
Report used in compatibility mode or goal mode Specify either **BOTH** or omit this option if the report is independent on the mode. This option has no effect on the background or Postprocesser session.

Except of **GATHER, REPORT, TITLE, DGTEXT, DRTEXT,** and **HELP**, all options are ignored, if the current picture is the second definition for a report with the same ID.

Example

The following example shows how to use the PICTURE macro to add a menu item to ERBFMENU. The menu item for the report is ANL, the data gatherer is ANLDG, the PF key is 23, the data reporter is ANLRP, the length of the relocate block is 32, the length of all data control sections is 0, and the report is a table report. The title of the report is USER ANALYSIS, the default operands for the gatherer and the reporter are 1,1,1. The maximum number of relocate blocks is 128.

```
ANLPIC PICTURE ID=ANL,GATHER=ANLDG,PFK=23,REPORT=ANLRP,RBLEN=32,FBLEN=0,
               RPTTYP=T,TITLE='USER ANALYSIS',DGTEXT='1,1,1',
               DRTEXT='1,1,1'MAXRBS=128
```

Separation of workload management modes

The separation of the two workload management modes in Monitor II is necessary to support new functionality for the Monitor II ISPF display session. For the other types of sessions it is not that relevant, for the background session it is even ignored.

Due to the separation, it is possible, to create a picture of one report that differs in processing or appearance between compatibility mode and goal mode. This means that it is possible to have different modules gather data or using different options in compatibility mode or goal mode, respectively.

The foreground processing routine calls that pair of gatherer and reporter that matches the current workload management mode at the time ENTER key is pressed. It is the responsibility of the respective gatherer or reporter module to detect a mode switch within the report built time. If the processing cannot be continued in such case, the gatherer or reporter is expected to return with RC=52.

The gatherer or reporter invoked is also responsible for verifying that the options passed to it still match the mode, unless they are independent. If RMF decided upon the mode at the time the ENTER key has been pressed that module *xxxx* has to be invoked, but after the decision the mode has changed and the module is not intended for the new mode, this module has to communicate this case to RMF, too. If the only valid mode for such options or module is compatibility mode, it returns with RC=44 to tell RMF that the report option is not applicable in goal mode. If the only valid mode for such options or module is goal mode, it returns with RC=56 to tell RMF that the report option is not applicable in compatibility mode.

Influence of WLMMODE option in the PICTURE macro

The WLMMODE option can be used to define two pictures for one and the same report: one for compatibility mode, the other for goal mode. If two pictures are specified for the same report, only **one** copy of SMF buffers is allocated. Thus, if the size of the buffers differ between the two modes, it is important to specify the parameters such that the bigger amount of storage is allocated. The type (table or row report) of this report cannot change from one mode to another. The only options that may differ are the ones listed in the description of WLMMODE.

If the WLMMODE option is omitted or if WLMMODE=BOTH is specified, only one picture is allowed to be specified for this report. Subsequent pictures with the same ID are ignored.

If WLMMODE=COMPAT is specified, a report is built only if the system is running in compatibility mode at the time the ENTER key is pressed. If the system is running in goal mode, the goal mode version of this report can be invoked only, if there is a picture definition with WLMMODE=GOAL, too. If the report cannot be invoked, because there is no version for the current mode available, RMF displays message ERBA031I.

If WLMMODE=GOAL is specified, the corresponding behavior appears.

TSO terminal user authorization

All the data collected and reported by RMF during a Monitor II TSO display session is obtained from commonly addressable storage that is not fetch protected. However, if your installation wants to limit the use of the command that starts an RMF Monitor II (RMFMON) session under TSO, one method available is to replace the RMF control section with your own module. For Monitor II you replace the control section ERBTSOCK. Your routine will then be invoked as part of the RMF response to the RMFMON command.

Note: You cannot protect the ISPF session by ERBTSOCK. Instead, RACF services should be used in order to prevent from unauthorized calling of RMF Monitor II.

ERBTSOCK (Monitor II) runs in problem state with a key of 8. When this control section gets control, register 1 points to a two-word address list. The first address points to the seven-byte userid of the user who has issued the RMFMON command. The second word points to the PSCB. Figure 4-14 illustrates the input parameter structure.

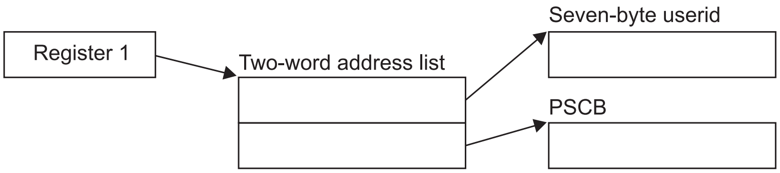


Figure 4-14. ERBTSOCK Input Parameter Structure

The module that you code to replace ERBTSOCK must be reenterable. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.

The processing your module performs depends on the method you choose to validate the user. Possible methods include issuing a RACHECK, prompting the user for a password, or checking the userid against a list of valid userids. Information on the TSO services available to perform these functions, such as TGET or TPUT, can be found in *z/OS TSO/E Programming Services*.

You can also use the PSCB bits defined for user use. This field (PSCBATR2 in the PSCB) comes from the UADS and can be updated by the USERDATA keyword of the ADD and CHANGE subcommands of the ACCOUNT command. See *z/OS TSO/E System Programming Command Reference* for more information on these commands.

TSO/E must be installed on your system to use the ACCOUNT, TGET and TPUT commands.

When your routine has completed its processing, set a return code of 0 in register 15 to indicate to RMF that the user is authorized to issue RMFMON. Set a return code of 4 in register 15 to indicate to RMF that the user is not authorized to issue RMFMON. In response to this return code, RMF displays a message to the terminal, and does not start the session. After setting the appropriate return code, RMF returns control by branching on the contents of register 14.

Mon II access control

For the Monitor II TSO/E display session the user authorization exit routine (ERBTSOCK) is part of the RMF load module that contains the RMFMON command. This module resides in SYS1.SERBLINK as load module RMFMON; its entry point is ERBMFTSO. Before your authorization routine can execute, you must link edit it with RMFMON; the control statements required are:

```
(ERBTSOCK object deck)
INCLUDE ddname(RMFMON)
ENTRY ERBMFTSO
NAME RMFMON(R)
```

End of Programming Interface information

Chapter 5. Adding Monitor III user exits

About Monitor III Exits and Reports

RMF provides user exits to allow you to tailor data collection and reporting to the needs of your installation. There are three main advantages to this. You can:

- Add information to a standard Monitor III report
- Sort the information in a standard report in a different order
- Create new reports combining the data that Monitor III gathers in the way you need them

In principle, you can modify any Monitor III report, with the exception of the Group Response Time report.

Overview

Programming Interface information

The **RMF Monitor III Utility** (see page 5-9) is the most important tool at your disposal for writing user exits. It is dialog-driven, and helps you use the necessary ISPF table services and RMF data-retrieval interface. However, you should be familiar with ISPF, Dialog Management Services, and RMF if you want to create and implement your own exit routines.

Data gathering

RMF generates Monitor III data by invoking a data gatherer module at each CYCLE. Replace the RMF dummy module ERB3GUSR with your own data gatherer routine, to have RMF invoke it, too, at each CYCLE.

Reporting

RMF takes several different actions in the course of producing a report, and the user exits allow you to modify each of these actions in order to change a report or produce a new one.

In the four separate processing **phases** of the reporter session, RMF:

1. Generates
2. Modifies
3. Formats and displays
4. Cleans up

the ISPF tables with the report data. The Monitor III Utility helps you to modify phases 1 and 3. Phases 2 and 4 are provided specially for user reports. See “Data reporter phases” on page 5-8 for more details.

Invoking user reports

The Monitor III Utility allows you to tailor RMF reports and to define the layout of new, user reports. RMF selects existing reports using ISPF SELECT, and uses the same method to select user-defined reports. To take advantage of this handling for your user reports:

- Use the Monitor III Utility to update the user-report selection panel
- Update the RMF command table, using the standard ISPF function

You can choose the time range to invoke the data reporter either:

- Before entering your user exit, by using the BREF/FREF commands or the RANGE/REFRESH session options
- Or from within the first phase of your reporter, by invoking the Data Retrieval Service module, ERB3RDRS, either by calling it or using the ISPF SELECT service.

“Data retrieval service (ERB3RDRS)” on page 5-36 describes this process.

Measurement data

The data gatherer collects data, and the data reporter uses this data to generate a formatted report for printing or display. The data gatherer module and the data reporter module communicate through control blocks that contain data from a set-of-samples.

Your user exits can use this means of communication, too. The format of the sample data is described in “Data gatherer sample structure”.

End of Programming Interface information

Data gatherer sample structure

Programming Interface information

RMF writes *resource data records* with the data that the gatherer routine collects at each CYCLE, and combines them into a *sample*. At the end of each MINTIME period, RMF combines these samples into a *set-of-samples* in the data gatherer’s address space, and moves the sets-of-samples into an in-storage buffer. The data reporter retrieves the data from this storage area, reduces it, and formats it for output.

Figure 5-1 shows the layout of three data areas that are common to all Monitor III data gatherers, whether coded by a user or provided by RMF. These areas are:

- The set-of-samples header
- The sample header
- The resource data record (RED)

Field offsets in the sample header and resource data record refer to offsets from the start of the control block containing the field. For example, the address of the first user record is the address of the REDG3 plus the offset to the first user record. All of these areas are maintained by RMF, specifically by the mainline data gathering module (ERB3GMFC). Figure 5-1 also shows the relationship between the data collected by the data gatherer user exit routine and the sample structure maintained by RMF.

Note: For a description of how Monitor III maintains a set of samples when VSAM data sets are used with data set support, see Chapter 6, “Using Monitor III VSAM data set support”, on page 6-1.

Sample structure

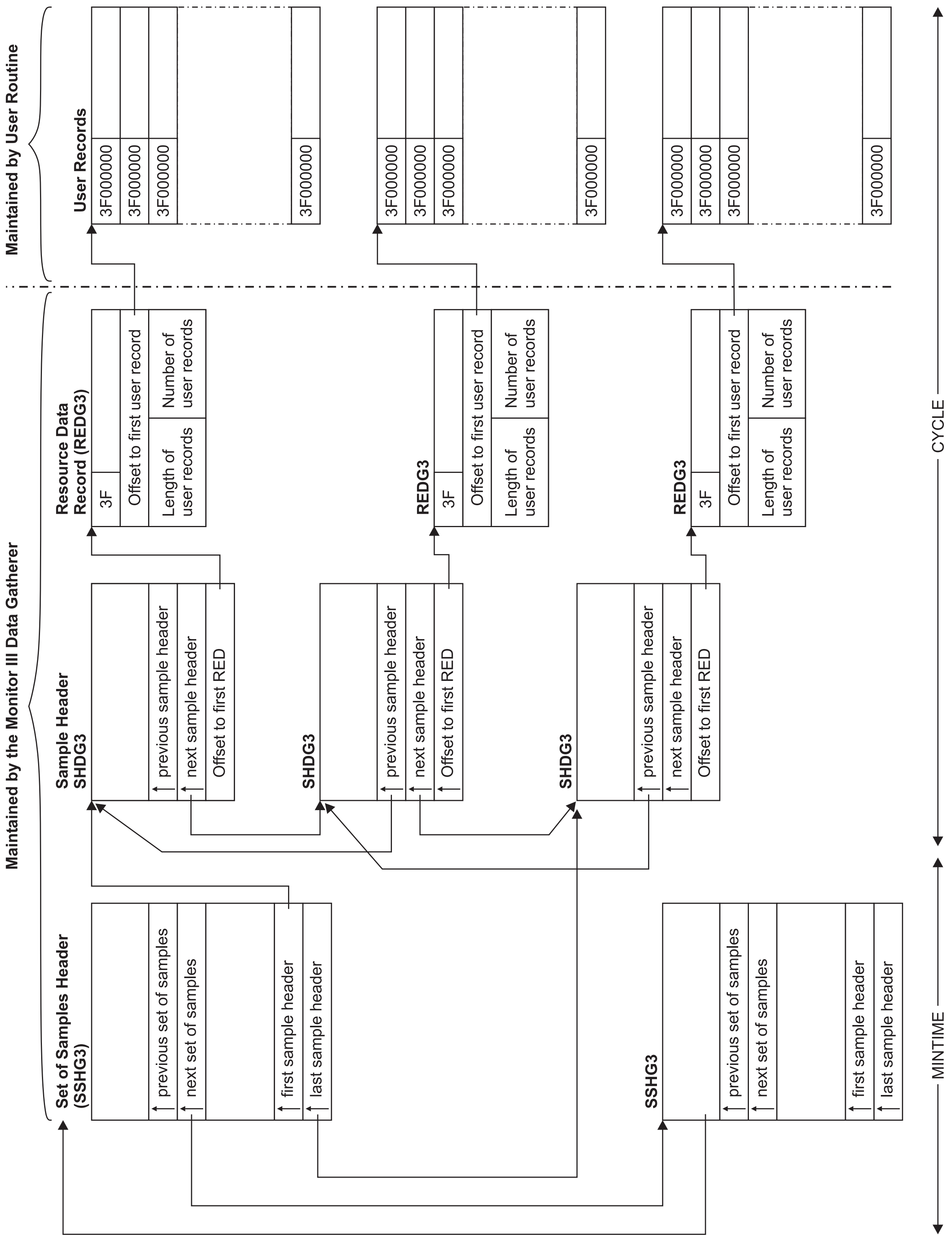


Figure 5-1. Data Gatherer Sample Structure

Data gatherer control blocks

Figure 5-2 describes the fields in the set-of-samples header control block, the sample header, and the resource data record. These data areas are mapped by the RMF macros ERBSSHG3, ERBSHDG3, and ERBREDG3.

```
                         SET OF SAMPLES HEADER
**********************  (ERBSSHG3 MAPPING MACRO)  ********************

SSHG3      DSECT           SAMPLE HEADER
           DS      0D      ALIGN ON DWORD BOUNDARY
SSHSSHG3   DS      XL5     ACRONYM SSHG3
SSHRMFV    DS      XL1     SSHG3 CONTROL BLOCK VERSION '05'X
SSHLEN     DS      H       LENGTH OF SSHG3
SSHRMFVN   DS      XL3     RMF VERSION NUMBER
SSHFLAG1   DS      XL1     FLAG BYTE
SSHGCOMP   EQU     X'80'   ON = DATA ARE COMPRESSED
SSHPREVP   DS      A       POINTER TO PREVIOUS SSH
SSHNEXTP   DS      A       POINTER TO NEXT SSH
           DS      4F      RESERVED
SSHSHDFP   DS      A       POINTER FIRST SAMPLE HEADER
SSHSHDLP   DS      A       POINTER TO LAST SAMPLE HEADER

                             SAMPLE HEADER
**********************  (ERBSHDG3 MAPPING MACRO)  ********************

SHDG3      DSECT           SAMPLE HEADER
           DS      0F      ALIGN ON WORD BOUNDARY
SHDSHDG3   DS      XL5     ACRONYM 'SHDG3'
SHDRMFV    DS      XL1     SHDG3 CONTROL BLOCK VERSION NUMBER X'02'
SHDLEN     DS      XL1     LENGTH OF SHDG3
SHDFLAG1   DS      XL1     SAMPLE FLAG 1
SHDINVAL   EQU     X'80'   SAMPLE IS INVALID
SHDPREVP   DS      A       POINTER TO PREVIOUS SAMPLE
SHDNEXTP   DS      A       POINTER TO NEXT SAMPLE
SHDREDOF   DS      A       OFFSET TO FIRST RED RECORD

                         RESOURCE DATA RECORD
**********************  (ERBREDG3 MAPPING MACRO)  ********************

REDG3      DSECT           RESOURCE RECORD
           DS      0F      ALIGN ON WORD BOUNDARY
REDREDID   DS      XL1     RED ID
REDUSRCB   EQU     X'3F'   RED ID FOR USER EXIT
REDFLAG1   DS      XL1     RED FLAG1
REDINVAL   EQU     X'80'   USER EXIT DATA ARE INVALID FOR THIS SAMPLE
REDRETRY   DS      H       NR OF RETRIES OF THE USER EXIT ROUTINE
REDFUWDO   DS      F       OFFSET TO FIRST USER EXIT RECORD
REDUSERL   DS      H       LENGTH OF USER EXIT RECORD
REDUSERN   DS      H       NUMBER OF USER EXIT RECORDS
```

Figure 5-2. Mapping Macros of ERBSSHG3, ERBSHDG3 and ERBREDG3

Set of samples header control block (SSHG3)

The set-of-samples header control block represents all samples collected during a MINTIME interval. This control block contains pointers to the previous and next set-of-samples header control block, as well as pointers to the first and last sample header control blocks. A set-of-samples is the smallest amount of data that the data reporter can retrieve. RMF maintains and updates all fields in this control block as needed.

Sample header control block (SHDG3)

This control block identifies a single sample taken at the end of a CYCLE. RMF identifies each sample with a sequence number and increments the sequence number at every CYCLE. This sample header contains forward and backward pointers to other sample header control blocks in the chain, as well as a pointer to the resource data record. RMF maintains and updates all fields in this control block as needed.

Resource data record (REDG3)

There is one resource data (RED) record for each defined resource in the system. RMF maintains and updates all fields in this record as needed. RMF uses RED records to access USE/WAIT records (in the case of the Monitor III data gatherer) or user records (in the case of a data gathering user exit routine). RED records are fixed in length, and contain X'3F' in the resource identifier (REDREDID) field when RMF invokes your data gatherer user exit routine. RMF uses this identifier to locate your user records, which also must have the same hexadecimal identifier. The RED record also contains the offset to the first user record (REDFUWDO), the length of your user exit records (REDUSERL), and the number of user exit records (REDUSERN) created during a CYCLE. While RMF maintains all the fields in the RED record, it obtains the length and number of user records from values you provide in the interface area used by the Monitor III data gatherer and your user routine. When RMF invokes your user exit, the second input parameter points to this interface area (see "Programming a data gatherer").

User record

A user record contains the information your data gathering routine collects at each CYCLE. The user record must be fixed in length and the first four bytes must contain the identifier X'3F000000'. You define the remaining fields in the user record and fill them in with the data you collect. The format of the data in the user record depends on the report you are generating. You set the format that best meets your needs.

End of Programming Interface information

Programming a data gatherer

Programming Interface information

The data gatherer runs in the Monitor III data gatherer address space in problem state, with a key of 8, and in 31-bit addressing mode. The data gatherer must be coded as reentrant. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. The register contents are:

Register 13	Address of the register save area
Register 14	Return address
Register 15	Entry address

Upon entry to the data gatherer, register 1 points to a contiguous list of three addresses that point to three input parameters. The first address points to the first parameter, the second address points to the second parameter, and the third address points to the third parameter. The input parameters are:

First Parameter

An area containing the management fields for the Monitor III data gatherer and the user data gatherer exit routine. The GGDMODAR DSECT (global data gatherer control block) is mapped by the ERBGGDG3 macro and describes the dynamic storage obtained when your data gatherer routine issues the GETMAIN macro. When RMF invokes your routine for the first time, it provides information in the following fields:

GGDMODNA
The module name, which is ERB3GUSR.

GGDAUSBP The subpool number from which your routine must obtain storage via the GETMAIN macro.

GGDREDID The resource identifier, which is X'3F'.

You must fill in the address and the length of the storage area (within the user subpool) that you obtain with the GETMAIN macro. The Monitor III data gatherer can then free this area at the end of the gatherer session. The fields in the global data gatherer control block that you must fill in are:
GGDAULEN The length of the storage area.
GGDAUPTR The address of the storage area.

All other fields in the GGDMODAR control block are set to zeroes. The contents of the fields in GGDMODAR are not changed by RMF between calls to your user exit routine.

Second Parameter
The interface area between the Monitor III data gatherer and the user exit routine. The interface area is reinitialized by RMF before each call to the exit. The interface area is four fullwords in length and contains the following:

- First fullword – The user subpool number from which the user exit routine must obtain storage via the GETMAIN macro if additional storage is required.
- Second fullword – The address of the retry work area (RETSTACK DSECT) used in error recovery. The ERBGGDG3 macro maps this retry work area. RMF provides this address, and your routine must not destroy it. The RETSTACK DSECT contains information that the Monitor III data gatherer error recovery module (ERB3GESA) uses if an error occurs in your data gatherer exit routine. Because RMF provides a recovery environment, it is not necessary to provide an ESTAE exit for your routine. If you choose to use the ESTAE or SPIE macro, you must not alter the Monitor III error recovery environment. You might choose to have your exit routine get control as a retry routine in the event of an abend. For example, if a control block chain changes while your data gatherer routine is scanning it, then your exit routine might abend. In this case, you must set up several fields in the retry work area at each invocation of your user exit routine, so that the Monitor III data gatherer can return control to your routine. These fields are:

  RETADDR Contains the retry entry point address in your routine. The data gatherer returns control to the user exit routine at this address when attempting to retry after an error. In cases where the number of retries is exhausted, the error recovery module (ERB3GESA) returns control to the main data gatherer module (ERB3GMFC) and not the data gatherer exit routine.

  RETCOUNT Contains the number of times the user exit routine can be retried during one invocation. The RMF error recovery routine decrements the number in this field each time it gets control.

  RETRUBFL Specifies registers that must be restored by the recovery termination manager (RTM) before returning control to the address in your routine specified in the RETADDR field.

This field should contain X‘FFFF’, indicating that all registers must be restored after must be restored after error-recovery processing completes.

RETREGSA A 16-word storage area used to store the contents of the registers specified in the RETRUBFL field.

- Third fullword – The address of an area containing the data the user exit routine collects. Your routine must supply this address each time it is invoked. RMF uses this address to move the collected data from the exit routine’s storage area into the data gatherer’s in-storage buffer.
- Fourth fullword – Two halfwords that the user exit routine must provide at each invocation. The first halfword must contain the length of the user record, and the second halfword must contain the number of user records collected during the current cycle. RMF places the length and number of user records in the resource data (RED) record. All user records must be fixed in length and must start with a fullword hexadecimal identifier of X‘3F000000’. RMF uses this information to move your collected data into the in-storage buffer.

Third Parameter

The address of the return code of the user exit routine.

If your user exit routine successfully gathers all the data needed for your report, set a return code of X‘00’ in the area pointed to by this parameter in the parameter list. RMF will invoke your user exit routine at the next CYCLE. If you do not want RMF to invoke your routine again, set a return code of X‘10’. Return to the caller by branching on the contents of register 14.

The processing your data gathering routine performs depends largely on the nature of the report for which you are gathering data. The first time RMF invokes your data gatherer routine, it provides a subpool number (in the GGDAUSBP field) that you must use when issuing the GETMAIN macro. After issuing a GETMAIN for the dynamic storage it needs to execute in, your routine must place the address and length of the storage obtained in the GGDAUPTR and GGDAULEN fields, respectively. (The GGDAUPTR and GGDAULEN fields contain zeroes when RMF invokes your routine for the first time.) When RMF makes subsequent calls to your routine, these two fields still contain the address and length of your dynamic storage. You do not have to issue another GETMAIN and you can reuse the storage obtained on the first call. This function eliminates the overhead of issuing a GETMAIN for dynamic storage each time RMF invokes your routine. Depending on the amount of data you collect, you may need to obtain additional storage to hold your user records.

End of Programming Interface information

Data reporter phases

Programming Interface information

To display a user-modified or user-created report, RMF makes use of ISPF tables that contain information about the report. You can control four phases to modify or create these tables and to generate and display your own reports for an RMF session.

Note: RMF uses two of these phases to generate and display standard RMF reports. Most of the unmodified standard reports, however, are not kept in ISPF tables. These tables are used primarily for user-modified and user-created reports.

The four phases and the activities performed in each are as follows:

- **Phase 1:** RMF generates an ISPF table that contains display data for every modifiable RMF report. Chapter 7, “Monitor III data reporter tables”, on page 7-1 describes these tables. The time range for the display data for your routine can be changed during this phase by calling the Data Retrieval Service (ERB3RDRS) module. See “Data retrieval service (ERB3RDRS)” on page 5-36 for information about how to invoke the Data Retrieval Service.

  RMF does not use the Data Retrieval Service.
- **Phase 2:** RMF invokes your routine to allow you to modify the ISPF table generated in phase 1 in order to change an existing report or create a new report. RMF does not use this phase; you supply your own routine.
- **Phase 3:** RMF formats the ISPF table created in phase 1 or modified in phase 2 and displays the tabular or graphic verion of the erport through the ISPF service TBDISPL.
- **Phase 4:** RMF invokes your routine to allow you to perform various clean-up operations (for example, to free resources allocated for use in previous phases). RMF does not use this phase; you supply your own routine.

Note: If you decide to replace any of these phases, you must conform to the standards and externals described in this manual. If you do not, the results are unpredictable. See “Installing your own phases” on page 5-32.

End of Programming Interface information

The Monitor III utility

Programming Interface information

To help you with the steps outlined above, use the Monitor III report format definition utility. This utility consists of a series of ISPF panels that allow you to modify the ISPF tables that RMF uses during the four phases.

The three ISPF tables used to control RMF report formatting and display are:

- The phase driver table ERBPHDS3, which contains all RMF-supplied report definitions to generate reports during phase 1.
- The tabular report format table ERBFMTS3, which contains the information used to format each RMF tabular report during phase 3.
- The graphic parameter report table ERBPTGS3, which contains entries for the graphic version of each RMF report during phase 3.

Chapter 7, “Monitor III data reporter tables”, on page 7-1 contains samples of each table and its entries.

You should be familiar with ISPF and TSO to use the report panel definition utility.

Report utility panel flow

Figure 5-3 shows the panel sequence for the report format definition utility.

To exit any panel, you can enter CANCEL on the command line or press END (PF3). If you enter CANCEL, the report format definition utility displays the report definition initialization panel (ERB3RD1) but saves none of your changes. If you press END on any panel, RMF displays the previous panel but does not save changes you have made. To continue viewing panels in sequence, press ENTER.

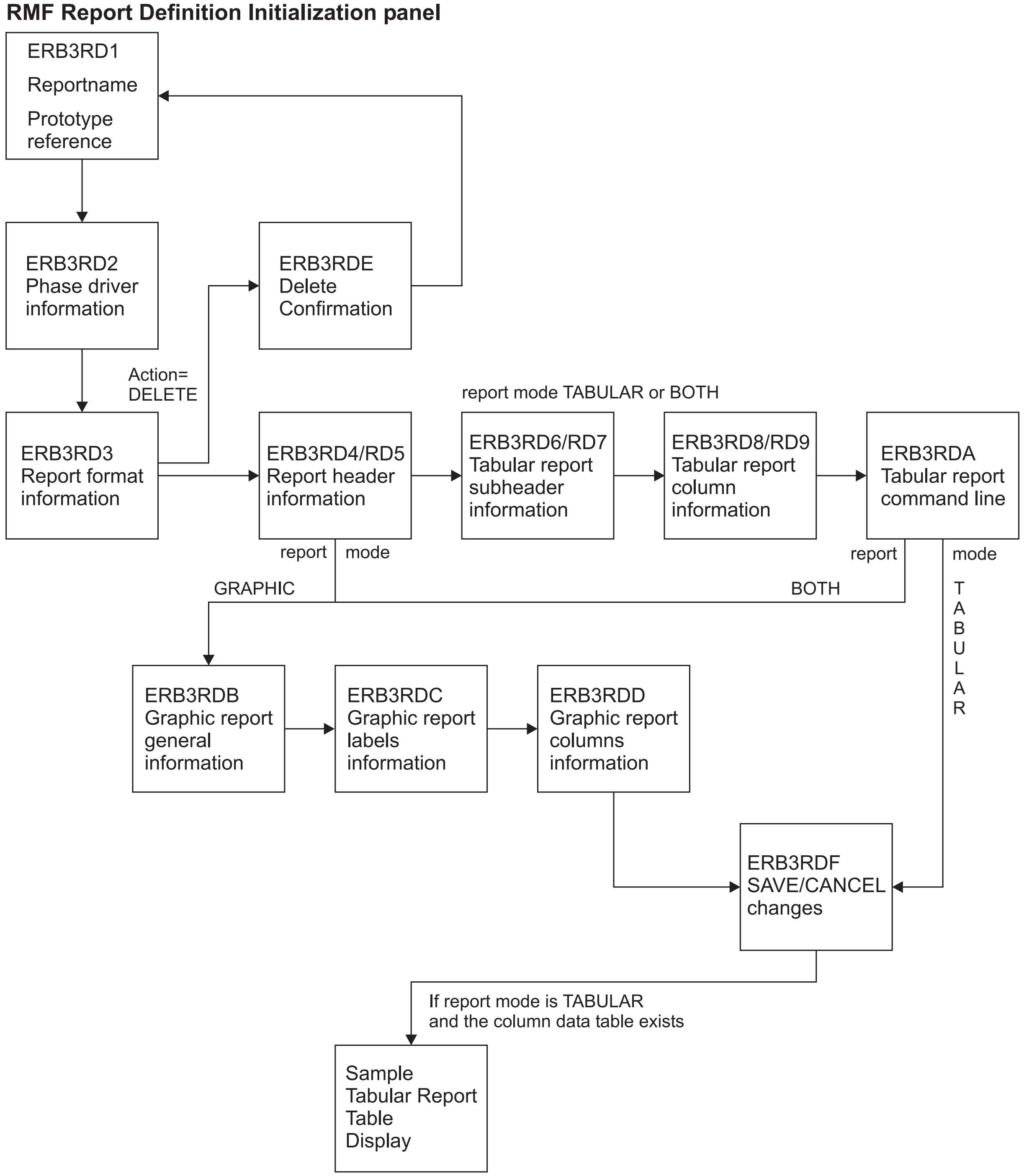

RMF Report Definition Initialization panel
ERB3RD1
Reportname
Prototype reference
ERB3RD2
Phase driver information
ERB3RDE
Delete Confirmation
Action= DELETE
ERB3RD3
Report format information
ERB3RD4/RD5
Report header information
report mode TABULAR or BOTH
ERB3RD6/RD7
Tabular report subheader information
ERB3RD8/RD9
Tabular report column information
ERB3RDA
Tabular report command line
report
mode
GRAPHIC
report
mode
BOTH
TABULAR
ERB3RDB
Graphic report general information
ERB3RDC
Graphic report labels information
ERB3RDD
Graphic report columns information
ERB3RDF
SAVE/CANCEL changes
If report mode is TABULAR and the column data table exists
Sample Tabular Report Table Display

Figure 5-3. Panel Sequence for the Report Definition Utility

Before you start the utility

Do not use the RMF distribution table library as your ISPF output library (ERBTAB); you could destroy standard RMF report formats as a result. Allocate

ERBTAB as part of a private user table library. You can concatenate this private library to the beginning of the RMF input table library (ERBTLIB) and can safely delete the ISPF tables you have modified or created (ERBPHDS3, ERBFMTS3, and ERBPTGS3) for your own reports.

You can merge your own libraries with RMF libraries. If you want to change the data set names and the allocations, modify CLIST ERBRMF3X. CLIST ERBRMF3X allocates the RMF ISPF libraries from the following distribution libraries:
- Panels from SYS1.SERBPENU
- Tables from SYS1.SERBTENU
- Messages from SYS1.SERBMENU

These CLISTs are available in SYS1.SERBCLS, which must be concatenated to your SYSPROC library.

Starting the report utility

To start the report format definition utility, enter either from TSO/E ready mode or within ISPF:

```
RMF UTIL
```

If you have the Kanji version of RMF, you start the Monitor III utility by entering:

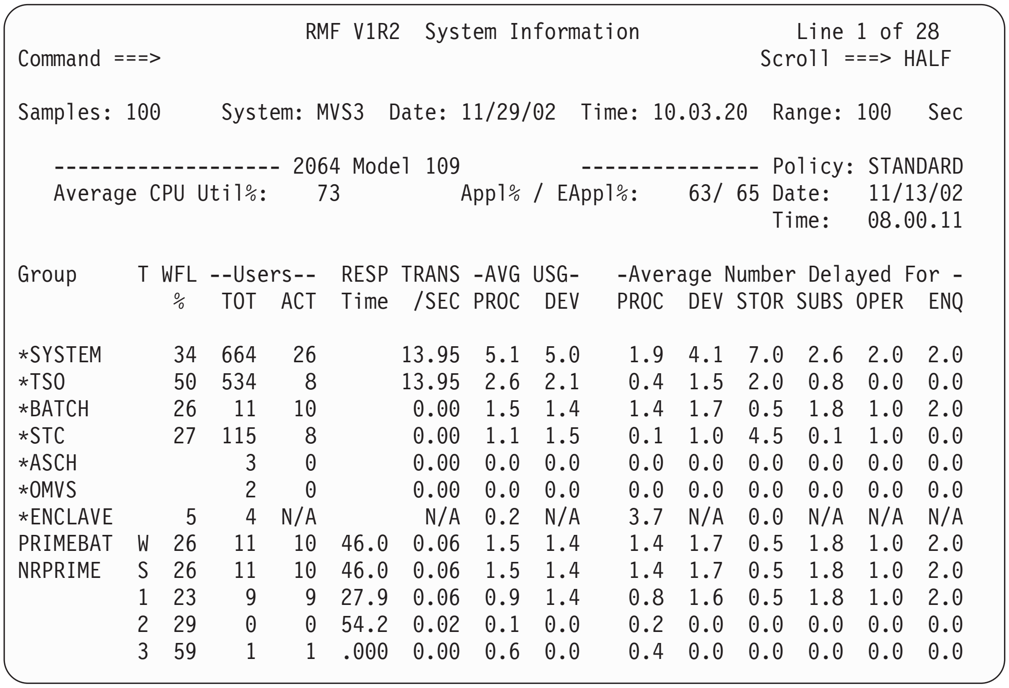
```
RMFJPN UTIL
```

Note: Do not try to access the report format definition utility in split screen mode when you are in an active RMF Monitor III reporter session.

For more information about a specific panel, use the HELP keys.

Example - Modified SYSINFO report

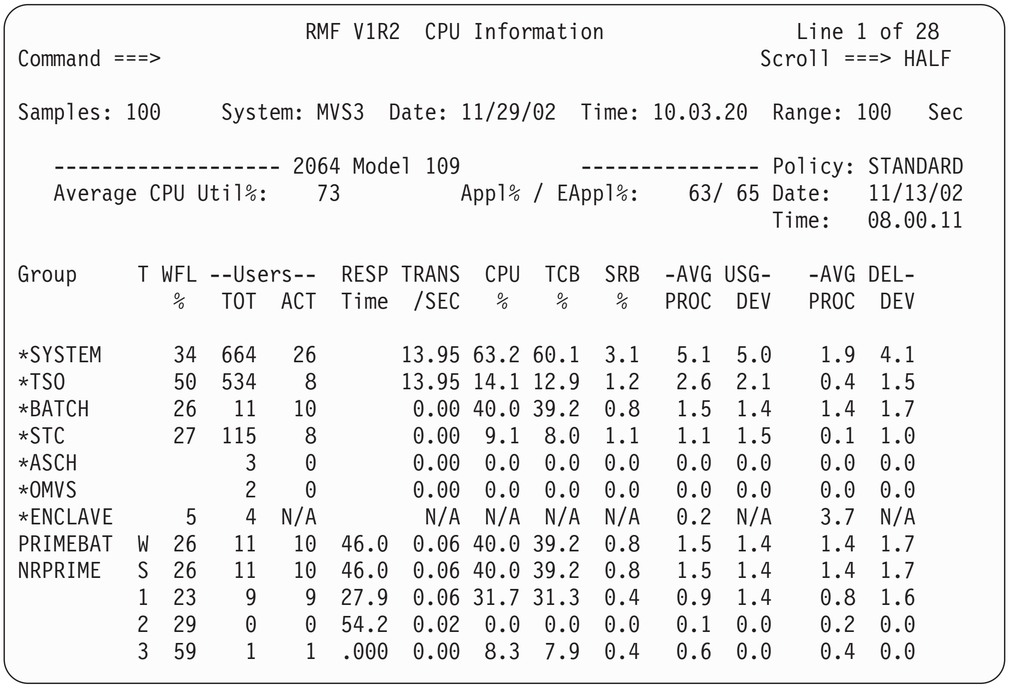
The task how to create a new Monitor III report will be shown based on the example of a modified SYSINFO report. The SYSINFO report has this format:



```
                         RMF V1R2  System Information          Line 1 of 28
Command ===>                                                Scroll ===> HALF

Samples: 100     System: MVS3  Date: 11/29/02  Time: 10.03.20  Range: 100   Sec

   ------------------- 2064 Model 109         --------------- Policy: STANDARD
   Average CPU Util%:    73          Appl% / EAppl%:    63/ 65 Date:   11/13/02
                                                               Time:   08.00.11

Group      T WFL --Users--  RESP TRANS -AVG USG-   -Average Number Delayed For -
             %   TOT  ACT   Time  /SEC PROC  DEV   PROC  DEV STOR SUBS OPER  ENQ

*SYSTEM      34  664   26        13.95  5.1  5.0    1.9  4.1  7.0  2.6  2.0  2.0
*TSO         50  534    8        13.95  2.6  2.1    0.4  1.5  2.0  0.8  0.0  0.0
*BATCH       26   11   10         0.00  1.5  1.4    1.4  1.7  0.5  1.8  1.0  2.0
*STC         27  115    8         0.00  1.1  1.5    0.1  1.0  4.5  0.1  1.0  0.0
*ASCH              3    0         0.00  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
*OMVS              2    0         0.00  0.0  0.0    0.0  0.0  0.0  0.0  0.0  0.0
*ENCLAVE      5    4  N/A          N/A  0.2  N/A    3.7  N/A  0.0  N/A  N/A  N/A
PRIMEBAT   W  26   11   10  46.0  0.06  1.5  1.4    1.4  1.7  0.5  1.8  1.0  2.0
NRPRIME    S  26   11   10  46.0  0.06  1.5  1.4    1.4  1.7  0.5  1.8  1.0  2.0
           1  23    9    9  27.9  0.06  0.9  1.4    0.8  1.6  0.5  1.8  1.0  2.0
           2  29    0    0  54.2  0.02  0.1  0.0    0.2  0.0  0.0  0.0  0.0  0.0
           3  59    1    1  .000  0.00  0.6  0.0    0.4  0.0  0.0  0.0  0.0  0.0
```

Figure 5-4. SYSINFO Report

The target is to create a report called SYSCPU that provides some more CPU related information as TCB% and SRB% for each group. This data is available in the corresponding Monitor III table.



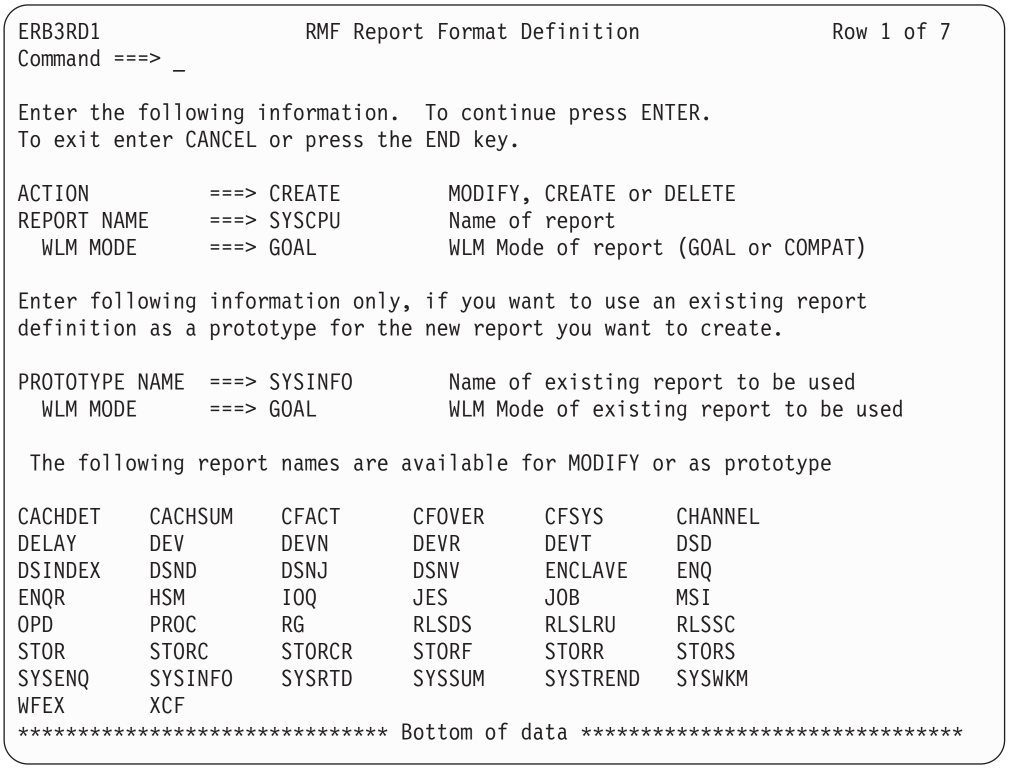
```
                         RMF V1R2  CPU Information             Line 1 of 28
Command ===>                                                Scroll ===> HALF

Samples: 100     System: MVS3  Date: 11/29/02  Time: 10.03.20  Range: 100   Sec

   ------------------- 2064 Model 109         --------------- Policy: STANDARD
   Average CPU Util%:    73          Appl% / EAppl%:    63/ 65 Date:   11/13/02
                                                               Time:   08.00.11

Group      T WFL --Users--  RESP TRANS  CPU  TCB  SRB  -AVG USG-   -AVG DEL-
             %   TOT  ACT   Time  /SEC   %    %    %   PROC  DEV   PROC  DEV

*SYSTEM      34  664   26        13.95 63.2 60.1  3.1   5.1  5.0    1.9  4.1
*TSO         50  534    8        13.95 14.1 12.9  1.2   2.6  2.1    0.4  1.5
*BATCH       26   11   10         0.00 40.0 39.2  0.8   1.5  1.4    1.4  1.7
*STC         27  115    8         0.00  9.1  8.0  1.1   1.1  1.5    0.1  1.0
*ASCH              3    0         0.00  0.0  0.0  0.0   0.0  0.0    0.0  0.0
*OMVS              2    0         0.00  0.0  0.0  0.0   0.0  0.0    0.0  0.0
*ENCLAVE      5    4  N/A          N/A  N/A  N/A  N/A   0.2  N/A    3.7  N/A
PRIMEBAT   W  26   11   10  46.0  0.06 40.0 39.2  0.8   1.5  1.4    1.4  1.7
NRPRIME    S  26   11   10  46.0  0.06 40.0 39.2  0.8   1.5  1.4    1.4  1.7
           1  23    9    9  27.9  0.06 31.7 31.3  0.4   0.9  1.4    0.8  1.6
           2  29    0    0  54.2  0.02  0.0  0.0  0.0   0.1  0.0    0.2  0.0
           3  59    1    1  .000  0.00  8.3  7.9  0.4   0.6  0.0    0.4  0.0
```

Figure 5-5. SYSCPU Report as Modification of the SYSINFO Report

You find details about all values than can be displayed for all Monitor III reports in Chapter 7, “Monitor III data reporter tables”, on page 7-1.

Report format definition panel (ERB3RD1)

After you call the report format definition utility by RMF UTIL, you get the **Report Definition Initialization** panel (ERB3RD1). On this panel, you can specify whether you want to create a new report or modify or delete an existing one. You can also select the name of an existing RMF report to use as a prototype for the new report.



```
ERB3RD1                 RMF Report Format Definition                Row 1 of 7
Command ===> _

Enter the following information.  To continue press ENTER.
To exit enter CANCEL or press the END key.

ACTION          ===> CREATE         MODIFY, CREATE or DELETE
REPORT NAME     ===> SYSCPU         Name of report
  WLM MODE      ===> GOAL           WLM Mode of report (GOAL or COMPAT)

Enter following information only, if you want to use an existing report
definition as a prototype for the new report you want to create.

PROTOTYPE NAME  ===> SYSINFO        Name of existing report to be used
  WLM MODE      ===> GOAL           WLM Mode of existing report to be used

 The following report names are available for MODIFY or as prototype

CACHDET    CACHSUM    CFACT      CFOVER     CFSYS      CHANNEL
DELAY      DEV        DEVN       DEVR       DEVT       DSD
DSINDEX    DSND       DSNJ       DSNV       ENCLAVE    ENQ
ENQR       HSM        IOQ        JES        JOB        MSI
OPD        PROC       RG         RLSDS      RLSLRU     RLSSC
STOR       STORC      STORCR     STORF      STORR      STORS
SYSENQ     SYSINFO    SYSRTD     SYSSUM     SYSTREND   SYSWKM
WFEX       XCF
******************************* Bottom of data ********************************
```

Figure 5-6. Report Definition Initialization Panel ERB3RD1

The panel fields and their meanings are:

ACTION
Specifies the action you want RMF to perform as follows:
MODIFY - to change an existing RMF report
CREATE - to create a new report
DELETE - to delete an existing report

REPORT NAME
Specifies the name of the report that RMF is to modify, create, or delete. The report name must conform to ISPF naming conventions.

WLM MODE
Specifies the mode of the report, either compatibility or goal mode.

PROTOTYPE NAME
When you enter CREATE for ACTION, specifies the name of an existing RMF report to use as a prototype or model for your report. RMF provides you those report values, which you can change when you modify or create your report.

When you enter MODIFY or DELETE for ACTION, you can ignore this field.

Phase driver information panel (ERB3RD2)

Press ENTER to display the next panel, the **Phase Driver Information** panel (ERB3RD2).

Phase driver

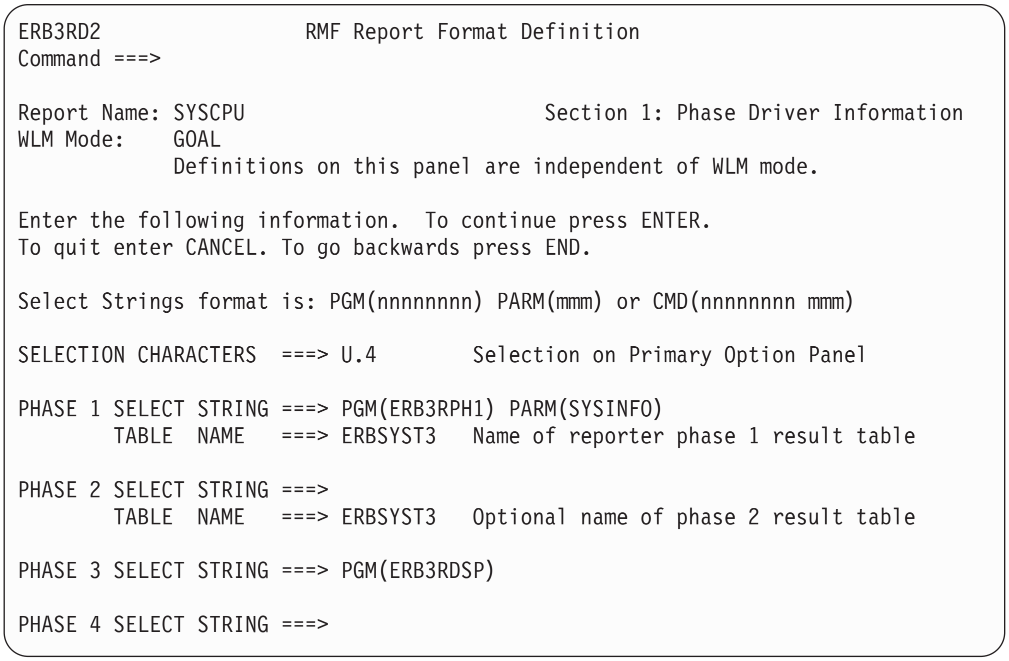
On this panel, you can specify the selection character(s) to use for the new or modified report on the Primary menu of a report session. You can also specify for each reporter phase the program or CLIST to modify, create, or print your report, or perform clean-up services and routines.

If you want to modify an existing RMF report without changing the layout or header information, you can provide your own program or CLIST for phase 2 on this panel. You can use ISPF services and commands like TBSORT, TBDELETE, or TBCREATE to perform these modifications during phase 2.

If you want to modify an existing RMF report format or layout without adding or deleting lines from a report, you can specify the name of the RMF report you want to modify for phase 1 (optionally for phase 2) and the name of the standard program that RMF uses to format RMF reports for phase 3. See PHASE 3 STRING in Figure 5-7. You can then use the remaining report format definition utility panels to make the header and layout changes for the modified report.

If you want to create a report, you should use a prototype (see Figure 5-6 for the Report Format Definition panel) and make sure to include the report selection on the Primary menu for the RMF report session.

Figure 5-7 is an example of a Phase Driver Information panel that contains information about the SYSINFO report. It assumes that the new SYSCPU report will become available as option 4 in the User Selection menu.



```
ERB3RD2                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                         Section 1: Phase Driver Information
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

Select Strings format is: PGM(nnnnnnnn) PARM(mmm) or CMD(nnnnnnnn mmm)

SELECTION CHARACTERS  ===> U.4       Selection on Primary Option Panel

PHASE 1 SELECT STRING ===> PGM(ERB3RPH1) PARM(SYSINFO)
        TABLE  NAME   ===> ERBSYST3   Name of reporter phase 1 result table

PHASE 2 SELECT STRING ===>
        TABLE  NAME   ===> ERBSYST3   Optional name of phase 2 result table

PHASE 3 SELECT STRING ===> PGM(ERB3RDSP)

PHASE 4 SELECT STRING ===>
```

Figure 5-7. Phase Driver Information Panel (ERB3RD2)

The panel fields and their meanings are as follows:

SELECTION CHARACTERS
Specifies a 1 to 8 character alphanumeric value that RMF uses as a selection value on the Primary menu of a report session. You must have defined these selection characters in the menu panel.

If you enter a selection that is currently used on the Primary menu of a report session, RMF displays the report that you modify or create on this panel when you make the selection.

PHASE 1 SELECT STRING
Specifies the name of the program or CLIST that the reporter control module (ERB3RDPC) uses to generate the ISPF report table during phase 1. You must specify a CLIST for CMD or program for PGM. (Follow the rules for ISPF SELECT services.) If you are modifying an existing RMF report or creating a new report using a prototype, you must specify for PGM the program name ERBRPH1, and for PARM the command name of the RMF report that you are modifying or using as a prototype. If you are creating a new report, be sure to include the report as a selection on the Primary menu or on the User Selection menu.

See the RMF supplied phase driver table (ERBPHDS3) in Chapter 7, “Monitor III data reporter tables”, on page 7-1 for a list of the RMF program and PARM names.

PHASE 1 TABLE NAME
Specifies the name of the ISPF table that results when your program or CLIST is invoked during phase 1. You must specify this parameter if you have specified PHASE 1 SELECT STRING.

For a list of the RMF report data tables (PHDRTAB1) in the RMF supplied phase driver table (ERBPHDS3), see Chapter 7, “Monitor III data reporter tables”, on page 7-1.

PHASE 2 SELECT STRING
Specifies the name of the program or CLIST used to modify the ISPF report data table created in phase 1. If you are creating a new report without having specified a prototype, you must enter the name of your CLIST to create the new report. (Follow the rules for ISPF SELECT services.) If you are modifying only the report header or layout of an existing RMF report, you do not need to enter a PHASE 2 SELECT STRING.

PHASE 2 TABLE NAME
Specifies the name of the ISPF table that results after phase 2. If you have entered a value for PHASE 2 SELECT STRING, you must specify a valid phase 2 table name.

If you are modifying the report header or layout of an existing RMF report, you can enter the same name you entered for PHASE 1 TABLE NAME.

PHASE 3 SELECT STRING
Specifies the program or CLIST that RMF uses to initiate phase 3 to format your report.

If you do not provide a program or CLIST for this field, RMF skips the remaining report format definition utility panels and displays the report definition initialization panel ERB3RD1. When you invoke your report during an RMF session, RMF does not display the report.

If you are creating a report and you want RMF to display it, specify PGM(ERB3RDSP), the standard RMF display module.

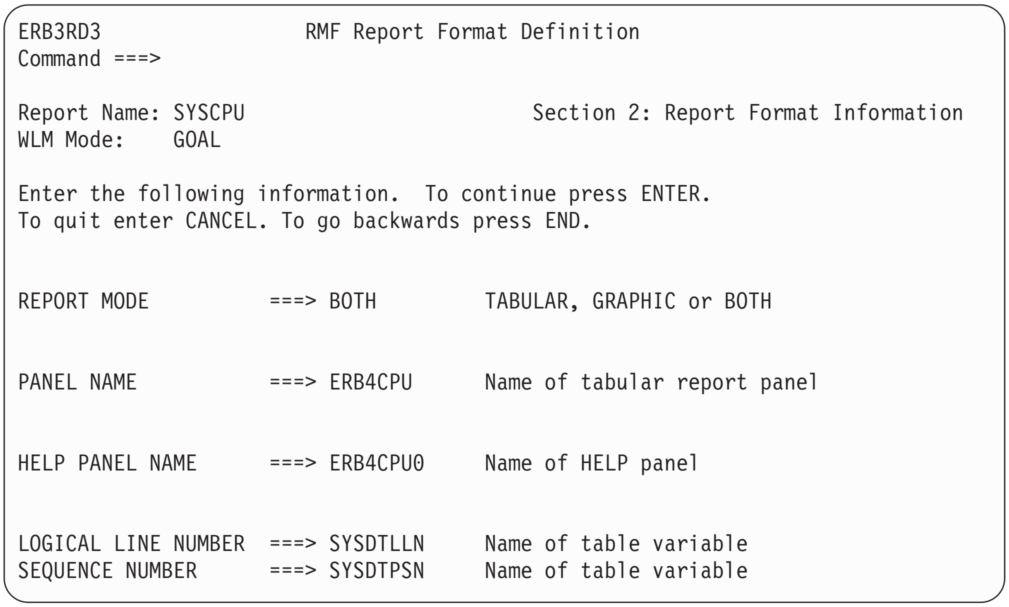
PHASE 4 SELECT STRING
Specifies the program or CLIST that ERB3RDPC uses to initiate phase 4. This field is optional.

Report format information panel (ERB3RD3)

If you have entered a name for PHASE 3 SELECT STRING on ERB3RD2, RMF next displays the **Report Format Information** panel (ERB3RD3). This panel is the first in a series of panels that allows you to change the header and subheader layout of an RMF report.

On this panel (ERB3RD3), you can specify tabular or graphic, or both the tabular and graphic displays for the report, the panel name of the tabular version of the report, or specify the name of a report help panel.

Figure 5-8 is an example of a Report Format Information panel for the SYSINFO report:



```
ERB3RD3                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                         Section 2: Report Format Information
WLM Mode:    GOAL

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.


REPORT MODE          ===> BOTH         TABULAR, GRAPHIC or BOTH


PANEL NAME           ===> ERB4CPU      Name of tabular report panel


HELP PANEL NAME      ===> ERB4CPU0     Name of HELP panel


LOGICAL LINE NUMBER  ===> SYSDTLLN     Name of table variable
SEQUENCE NUMBER      ===> SYSDTPSN     Name of table variable
```

Figure 5-8. Report Format Information Panel (ERB3RD3)

The panel fields and their meanings are as follows:

REPORT MODE
Specifies the display mode for the report. Valid values are as follows:
TABULAR
GRAPHIC
BOTH

PANEL NAME
Specifies the name of the ISPF display panel for the tabular version of the report when you enter TABULAR or BOTH for REPORT MODE.

For a tabular report, you must specify the name of the display panel that is to contain the report information. RMF-supplied panel names that you can use are ERB3DSI (if you are modifying or using the DI screen as a prototype), ERB3SRR (if you are modifying or using the STORR delay report as a prototype), ERB3SYS (if you are modifying or using the SYSINFO report as a prototype), ERB3WFX (if you are modifying or using the WFEX report as a prototype), or ERB3CMN (if you are modifying or using any other report as a prototype).

If you specify the name of your own panel, make sure that the panel includes the following information:

- Output fields for 2 standard header lines (DSPHDR1 and DSPHDR2)
- Output fields for up to 5 subheader lines (DSPSUBH1 - DSPSUBH5) contained in the RMF report you want to modify
- Output fields for up to 3 column header lines (FMTCOLH1 -FMTCOLH3) contained in the RMF report you want to modify. For a description of the report format table ERBFMTS3, see Chapter 7, "Monitor III data reporter tables", on page 7-1.
- Up to 3 model line variables (FMTMODL1 - FMTMODL3) contained in the model section of the RMF report you want to modify. For a description of the entries in the report format table ERBFMTS3, see Chapter 7, "Monitor III data reporter tables", on page 7-1.
- The command line (defined by variable ZCMD) and scroll amount field (defined by variable AMT)

Also, ensure that the user-defined panel for your report includes an initialization (INIT), reinitialization (REINIT), and processing (PROC) section as in the RMF-supplied panels.

If you enter GRAPHIC for REPORT MODE, leave PANEL NAME blank.

HELP PANEL NAME
When you enter a value for PANEL NAME, specifies the name of the ISPF help panel that contains help information for your report. The field is optional.

LOGICAL LINE NUMBER/SEQUENCE NUMBER
Specifies the name of key variables in the data table of the RMF report you are modifying. A logical line number identifies a logical group of related data rows within a report; a line sequence number identifies each physical table row that belongs to the logical group.

The logical line number (that identifies the entire data group) is 1; the sequence number (the number of physical lines that belong to the logical group and include the volume serial/device type on one line and the space type on the second line of the graphic report) is 2 or more.

When you toggle between tabular and graphic reports, RMF uses these variables to synchronize the line or bar displayed on the screen (the beginning of a logical group of data table rows). For examples of RMF report data tables, see Chapter 7, "Monitor III data reporter tables", on page 7-1.

Report header layout panels (ERB3RD4 and ERB3RD5)

Press ENTER to display the next panel, ERB3RD4, the **Report Header Layout** panel.

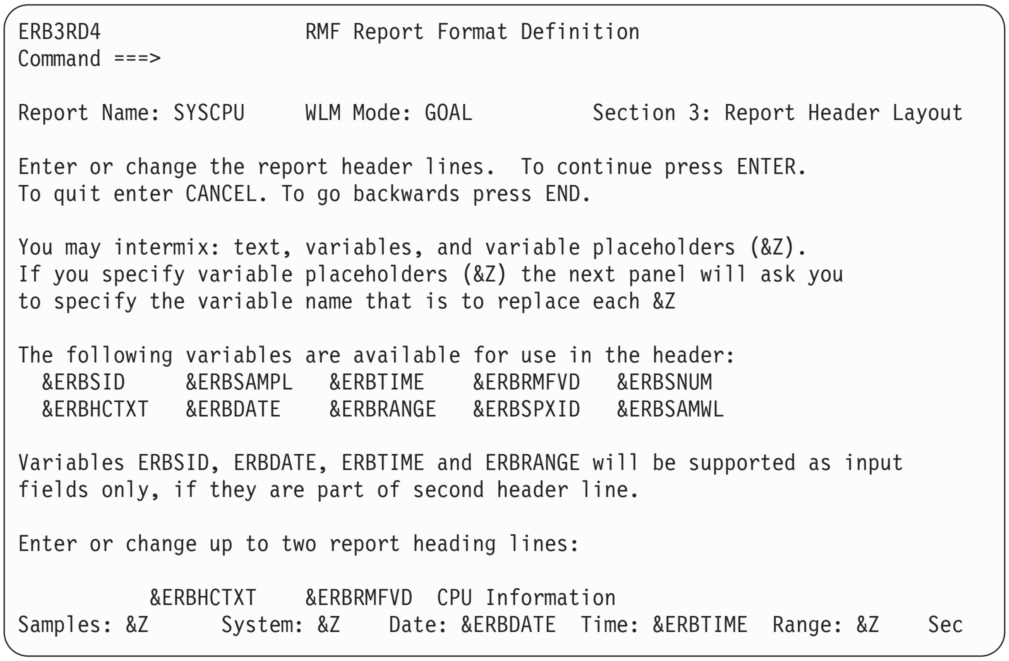
Each RMF report contains report headings, subheadings, and columns that you can modify. The Report Header Layout panels (ERB3RD4 and ERB3RD5) allow you to change up to 2 header lines for the tabular and graphic versions of the report.

On the first of these panels (ERB3RD4), you can specify the header lines and header variables for your report. At the bottom of the panel, enter the header lines exactly as you want them to appear on your report. You can use the variables listed on the panel to appear in the headings of your report. (Panel ERB3RD4 lists variables from header data table ERBHDRS3. For the meaning of all variables in ERBHDRS3, see Chapter 7, "Monitor III data reporter tables", on page 7-1.)

Header layout

If a variable name is too long to enter in the header line, you can use a placeholder (&Z). After you press ENTER, you define these placeholders with variable names on the next panel.

Figure 5-9 is an example of a report format definition panel ERB3RD4 that shows you the headings and variables for the SYSINFO report with the modified report title `CPU Information`:



```
ERB3RD4                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL          Section 3: Report Header Layout

Enter or change the report header lines.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).
If you specify variable placeholders (&Z) the next panel will ask you
to specify the variable name that is to replace each &Z

The following variables are available for use in the header:
  &ERBSID     &ERBSAMPL   &ERBTIME    &ERBRMFVD   &ERBSNUM
  &ERBHCTXT   &ERBDATE    &ERBRANGE   &ERBSPXID   &ERBSAMWL

Variables ERBSID, ERBDATE, ERBTIME and ERBRANGE will be supported as input
fields only, if they are part of second header line.

Enter or change up to two report heading lines:

           &ERBHCTXT    &ERBRMFVD  CPU Information
Samples: &Z      System: &Z    Date: &ERBDATE  Time: &ERBTIME  Range: &Z    Sec
```

Figure 5-9. Report Header Layout Panel (ERB3RD4)

In Figure 5-9, two report header lines appear at the bottom of the panel and ten variable names are available for the header lines.

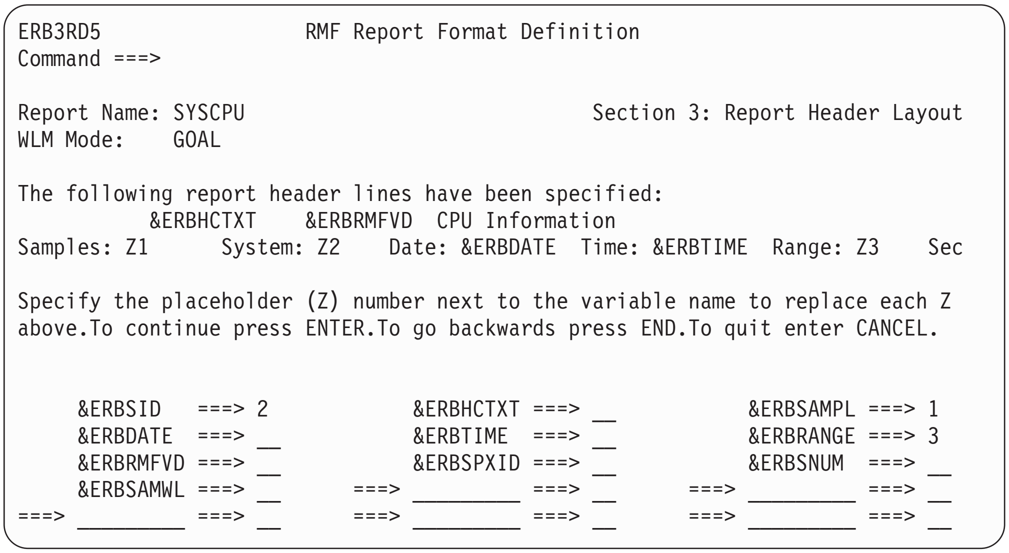
- Variables &ERBHCTXT and &ERBRMFVD are specified at the beginning of the first header line.
- Variables &ERBDATE and &ERBTIME are specified for `Date` and `Time`.
- Placeholders (&Z) for the other variables (&ERBSID for session id, &ERBSAMPL for samples, and &ERBRANGE for range) appear in the appropriate fields of the header lines and indicate that the variable names they represent might not fit in the space provided. These placeholders can be defined on the next panel.

Press ENTER to display the second **Report Header Layout** panel (ERB3RD5).

On ERB3RD5, you can specify variable names for any Z placeholders you have used. The headings, variables names, and placeholders as you entered them on ERB3RD4 appear at the top of the panel. The variable names appear under the headings in the order specified on ERB3RD4. You can specify your own variable names in the spaces provided; however, in order for RMF to display the user-specified variables during a report session, they must be in the function pool for phase 3 or in the shared ISPF variable pool. Otherwise, blanks appear in the report. See “Installing your own phases” on page 5-32.

You must specify a number for each Z placeholder and its corresponding variable. Numbers must start with 1 and continue in sequence. There must be a one-to-one correspondence between placeholders and variable names, each pair with a unique number assigned to indicate the order of placement of the variable.

Figure 5-10 is an example of Report Header Layout panel ERB3RD5 that defines the placeholders used on the previous panel. If you do not have placeholders to define, press ENTER to get the next panel.



```
ERB3RD5                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                             Section 3: Report Header Layout
WLM Mode:    GOAL

The following report header lines have been specified:
           &ERBHCTXT    &ERBRMFVD  CPU Information
Samples: Z1      System: Z2    Date: &ERBDATE  Time: &ERBTIME  Range: Z3    Sec

Specify the placeholder (Z) number next to the variable name to replace each Z
above.To continue press ENTER.To go backwards press END.To quit enter CANCEL.


     &ERBSID   ===> 2           &ERBHCTXT ===> __          &ERBSAMPL ===> 1
     &ERBDATE  ===> __          &ERBTIME  ===> __          &ERBRANGE ===> 3
     &ERBRMFVD ===> __          &ERBSPXID ===> __          &ERBSNUM  ===> __
     &ERBSAMWL ===> __      ===> _________ ===> __      ===> _________ ===> __
===> _________ ===> __      ===> _________ ===> __      ===> _________ ===> __
```

Figure 5-10. Report Header Layout Panel (ERB3RD5)

- Variable &ERBSAMPL that contains the number of samples replaces Z1.
- Variable &ERBSID that contains the session id replaces Z2 in the first header line of the report.
- &ERBRANGE that contains the range value replaces Z3 in the second header line.

Depending on your selection on panel ERB3RD3, you will continue as follows:

- If you specified TABULAR or BOTH for report mode, RMF displays the **Report Subheader Layout** panel ERB3RD6.
- If you specified GRAPHIC for report mode, RMF displays the **Graphic Parameter Definition** panel ERB3RDB, see "Graphic parameter definition panels (ERB3RDB, ERB3RDC, ERB3RDD)" on page 5-24.

Report subheader layout panels (ERB3RD6 and ERB3RD7)

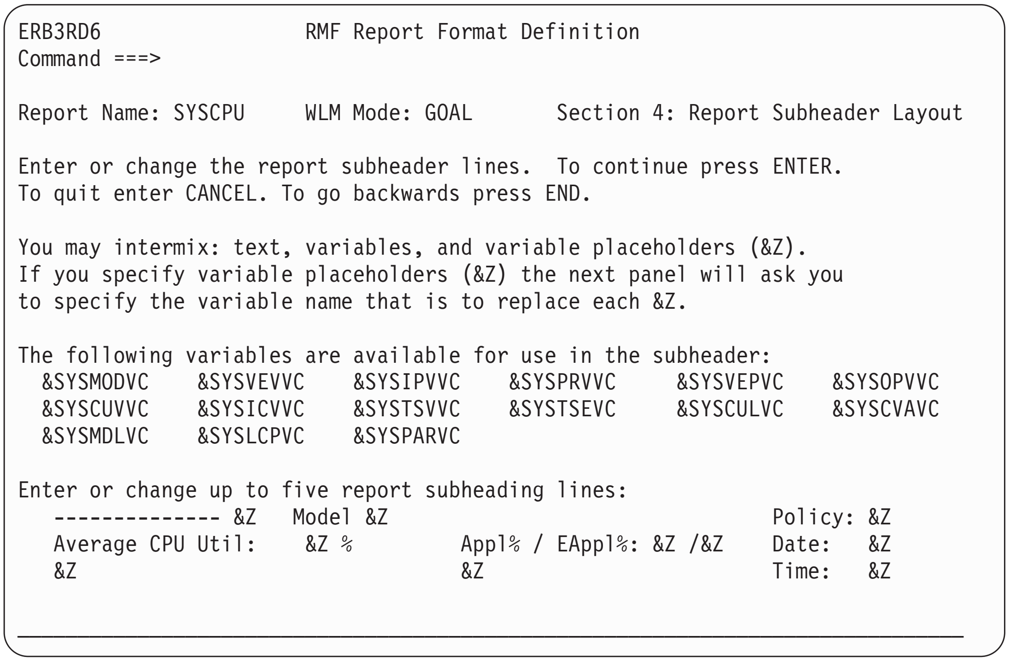
The Report Subheader Layout panel (ERB3RD6) displays up to five subheader lines of an existing RMF report. (Subheader lines are any lines in an RMF report that appear between the two standard header lines and the column headings.) ERB3RD6 also lists the variables that are available for use in the subheader lines of the modified report.

At the bottom of ERB3RD6, you enter the subheader lines exactly as you want them to appear on your report. You can use the variables listed on the panel to appear in the subheadings of your report. Panel ERB3RD6 lists variables from header data table ERBHDRS3.

If a variable name is too long to appear in the header line, you can use a placeholder (&Z). After you press ENTER, you define these placeholders with variable names on the next panel.

Subheader layout

Figure 5-11 is an example of a Report Subheader Layout panel ERB3RD6 that shows the subheadings of the SYSINFO report.



```
ERB3RD6                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL      Section 4: Report Subheader Layout

Enter or change the report subheader lines.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).
If you specify variable placeholders (&Z) the next panel will ask you
to specify the variable name that is to replace each &Z.

The following variables are available for use in the subheader:
  &SYSMODVC    &SYSVEVVC    &SYSIPVVC    &SYSPRVVC     &SYSVEPVC    &SYSOPVVC
  &SYSCUVVC    &SYSICVVC    &SYSTSVVC    &SYSTSEVC     &SYSCULVC    &SYSCVAVC
  &SYSMDLVC    &SYSLCPVC    &SYSPARVC

Enter or change up to five report subheading lines:
   -------------- &Z   Model &Z                                Policy: &Z
   Average CPU Util:    &Z %        Appl% / EAppl%: &Z /&Z    Date:   &Z
   &Z                               &Z                         Time:   &Z

________________________________________________________________________________
```

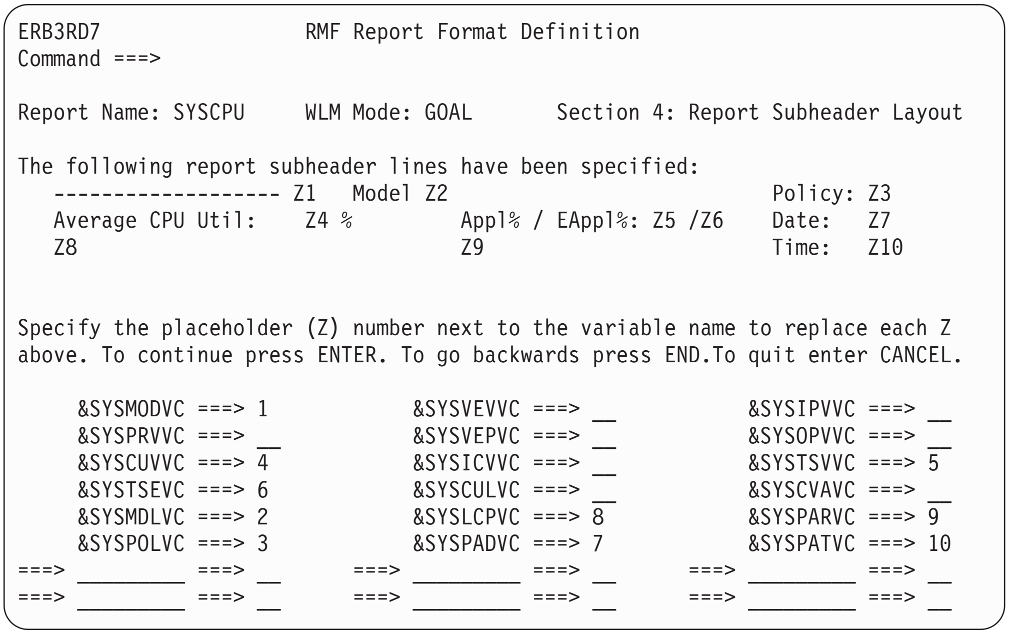
Figure 5-11. Report Subheader Layout Panel (ERB3RD6)

In Figure 5-11, subheader lines appear at the bottom of the panel and 14 variable names from the SYSINFO report are available. You can modify these subheader lines and indicate where you want the available variables to appear in them.

Press ENTER to display the next panel ERB3RD7, the second Report Subheader Layout panel.

On this panel, you can specify variable names for any Z placeholders you have used. For a description of how to replace placeholders with variable names, see the Report Header Layout panel (Figure 5-10).

Figure 5-12 shows panel ERB3RD7 that defines placeholders used on the previous panel.



```
ERB3RD7                RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL      Section 4: Report Subheader Layout

The following report subheader lines have been specified:
   ------------------- Z1   Model Z2                           Policy: Z3
   Average CPU Util:    Z4 %        Appl% / EAppl%: Z5 /Z6    Date:   Z7
   Z8                               Z9                         Time:   Z10


Specify the placeholder (Z) number next to the variable name to replace each Z
above. To continue press ENTER. To go backwards press END.To quit enter CANCEL.

     &SYSMODVC ===> 1           &SYSVEVVC ===> __          &SYSIPVVC ===> __
     &SYSPRVVC ===> __          &SYSVEPVC ===> __          &SYSOPVVC ===> __
     &SYSCUVVC ===> 4           &SYSICVVC ===> __          &SYSTSVVC ===> 5
     &SYSTSEVC ===> 6           &SYSCULVC ===> __          &SYSCVAVC ===> __
     &SYSMDLVC ===> 2           &SYSLCPVC ===> 8           &SYSPARVC ===> 9
     &SYSPOLVC ===> 3           &SYSPADVC ===> 7           &SYSPATVC ===> 10
===> _________ ===> __      ===> _________ ===> __      ===> _________ ===> __
===> _________ ===> __      ===> _________ ===> __      ===> _________ ===> __
```

Figure 5-12. Report Subheader Layout Panel (ERB3RD7)

Report column layout panels (ERB3RD8 and ERB3RD9)

Press ENTER to display the next panel, ERB3RD8, the report subheader first **Report Column Layout** panel.

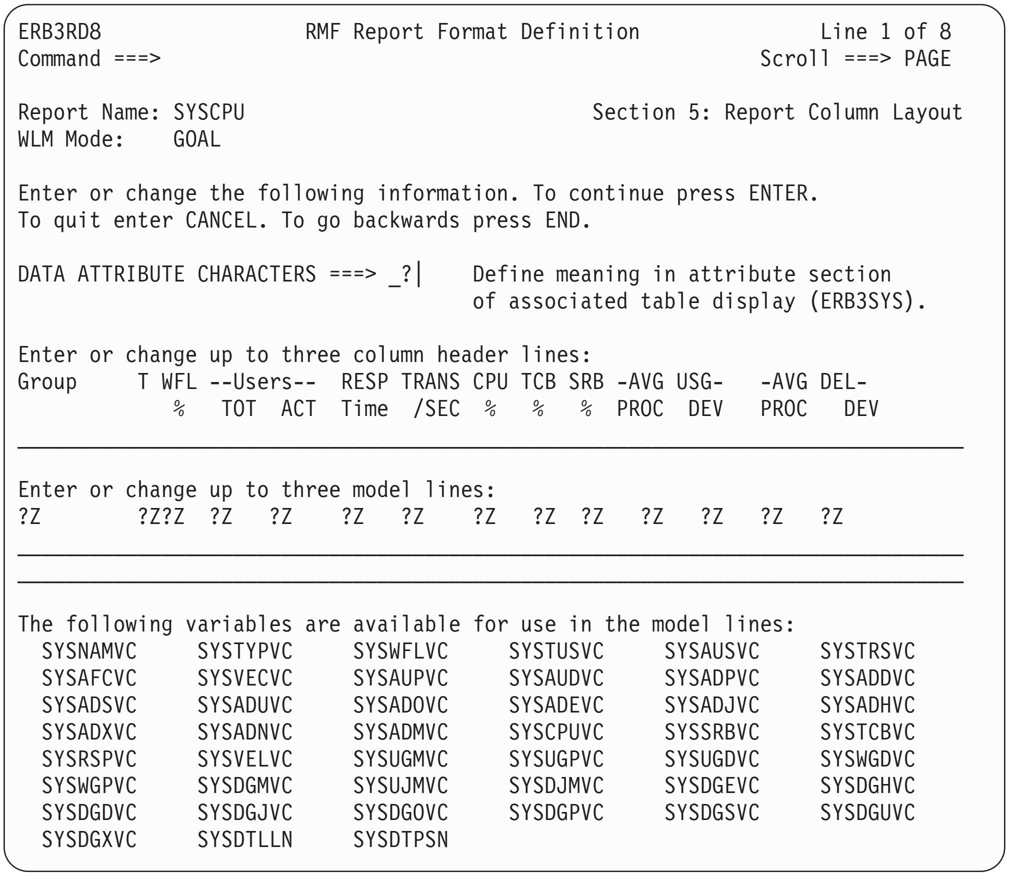
On this panel, you can modify report columns. You can enter up to three column header lines as you want them to appear in the report.

You can specify up to three model lines for your columns by using an attribute character followed by a variable name or placeholder (&Z). (See DATA ATTRIBUTE CHARACTERS described below.)

You can use the variable names listed at the bottom of the panel to appear in the columns of your report. This panel also allows you to specify a placeholder (&Z) for any variable name you want to use. (Panel ERB3RD8 lists variables from the data table of the RMF report you are modifying. All variables might not appear on the first page of the panel. Scroll through the panel and select the variable names you need. For information about RMF report data tables, see Chapter 7, “Monitor III data reporter tables”, on page 7-1.) You can define placeholders for variable names on the next panel.

Figure 5-13 is an example of ERB3RD8 that shows report column headings for the modified SYSINFO report with columns that contain data about TCB%, SRB%, and execution velocity. The details about delay percentages have been removed.

Column layout



```
ERB3RD8                RMF Report Format Definition               Line 1 of 8
Command ===>                                                    Scroll ===> PAGE

Report Name: SYSCPU                                   Section 5: Report Column Layout
WLM Mode:    GOAL

Enter or change the following information. To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

DATA ATTRIBUTE CHARACTERS ===> _?|    Define meaning in attribute section
                                      of associated table display (ERB3SYS).

Enter or change up to three column header lines:
Group     T WFL --Users--  RESP TRANS CPU TCB SRB -AVG USG-   -AVG DEL-
            %   TOT  ACT  Time  /SEC  %   %   %  PROC  DEV   PROC   DEV
_______________________________________________________________________________

Enter or change up to three model lines:
?Z        ?Z?Z  ?Z   ?Z    ?Z   ?Z    ?Z   ?Z  ?Z   ?Z   ?Z   ?Z   ?Z
_______________________________________________________________________________
_______________________________________________________________________________

The following variables are available for use in the model lines:
  SYSNAMVC     SYSTYPVC     SYSWFLVC     SYSTUSVC     SYSAUSVC     SYSTRSVC
  SYSAFCVC     SYSVECVC     SYSAUPVC     SYSAUDVC     SYSADPVC     SYSADDVC
  SYSADSVC     SYSADUVC     SYSADOVC     SYSADEVC     SYSADJVC     SYSADHVC
  SYSADXVC     SYSADNVC     SYSADMVC     SYSCPUVC     SYSSRBVC     SYSTCBVC
  SYSRSPVC     SYSVELVC     SYSUGMVC     SYSUGPVC     SYSUGDVC     SYSWGDVC
  SYSWGPVC     SYSDGMVC     SYSUJMVC     SYSDJMVC     SYSDGEVC     SYSDGHVC
  SYSDGDVC     SYSDGJVC     SYSDGOVC     SYSDGPVC     SYSDGSVC     SYSDGUVC
  SYSDGXVC     SYSDTLLN     SYSDTPSN
```

Figure 5-13. Report Column Layout Panel (ERB3RD8)

DATA ATTRIBUTE CHARACTERS

Specifies the ISPF characters used to indicate the start of a data field. Specify the data attribute characters before each variable name or placeholder (&Z) used in the model lines.

You must specify the name of a panel for the tabular version of a new or modified report. For RMF-supplied panels, the attribute characters appear as follows:

- a question mark (?) indicates that the output display characters appear unhighlighted (low intensity) in turquoise
- a slash (|) indicates that the output display characters appear highlighted (high intensity) in white
- a blank indicates that the input display characters appear unhighlighted (high intensity) in green

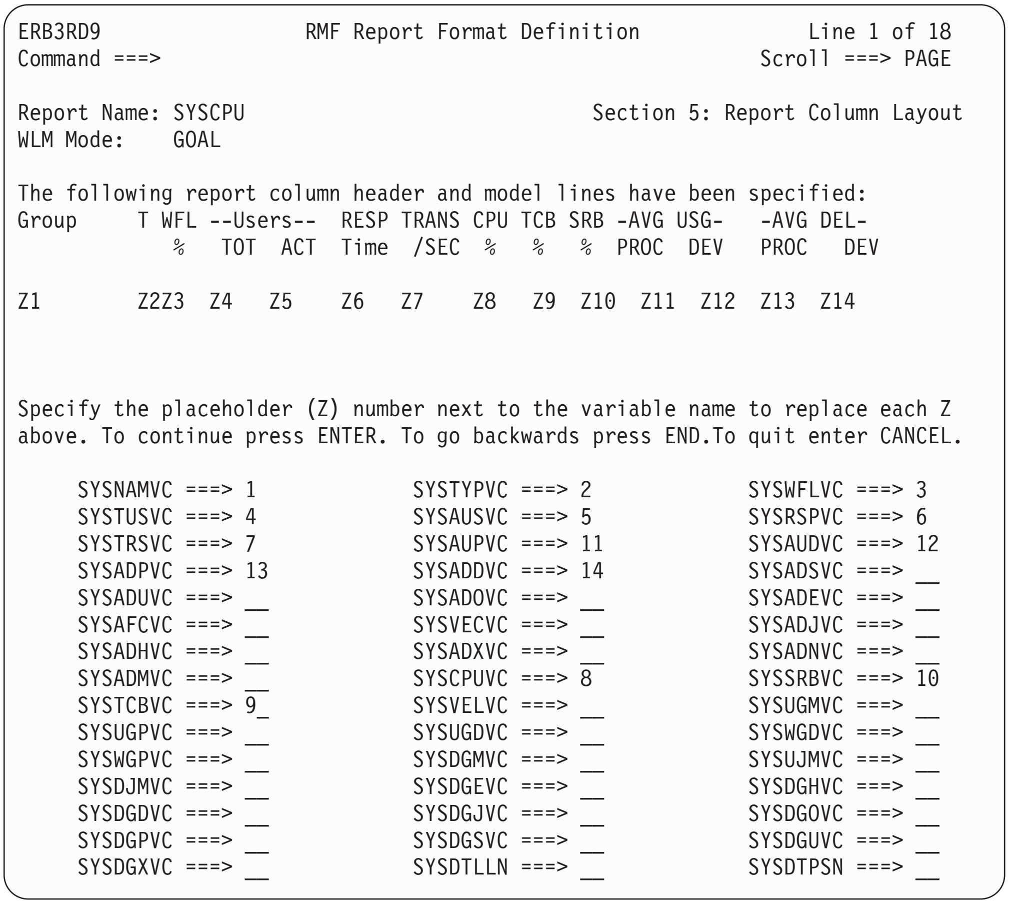
For user-defined panels, be sure that the data attribute characters match the characters in the attribute section of your ISPF display panel. See PANEL NAME on the report format information panel (ERB3RD3).

Press ENTER to display the next panel ERB3RD9, the second Report Column Layout panel.

On this panel, you can specify variable names for any Z placeholders you have used. The variable names available on the previous panel are listed at the bottom; you can add your own variable names in the spaces provided. If your variable names are not available when you invoke the report, blanks will appear instead of data. See the report header information panel (ERB3RD5) in Figure 5-10 for a description of how to replace placeholders with variable names.

If not all variable names appear on the first page of the panel, scroll through the remaining pages of the panel to see all available variable names.

Figure 5-14 is an example of Report Column Layout panel ERB3RD9 that defines placeholders used on the previous panel.



```
ERB3RD9                 RMF Report Format Definition              Line 1 of 18
Command ===>                                                  Scroll ===> PAGE

Report Name: SYSCPU                             Section 5: Report Column Layout
WLM Mode:    GOAL

The following report column header and model lines have been specified:
Group     T WFL --Users--  RESP TRANS CPU TCB SRB -AVG USG-   -AVG DEL-
             %   TOT  ACT  Time  /SEC  %   %   %  PROC  DEV   PROC   DEV

Z1        Z2Z3  Z4   Z5    Z6   Z7    Z8   Z9  Z10  Z11  Z12  Z13  Z14



Specify the placeholder (Z) number next to the variable name to replace each Z
above. To continue press ENTER. To go backwards press END.To quit enter CANCEL.

     SYSNAMVC ===> 1             SYSTYPVC ===> 2             SYSWFLVC ===> 3
     SYSTUSVC ===> 4             SYSAUSVC ===> 5             SYSRSPVC ===> 6
     SYSTRSVC ===> 7             SYSAUPVC ===> 11            SYSAUDVC ===> 12
     SYSADPVC ===> 13            SYSADDVC ===> 14            SYSADSVC ===> __
     SYSADUVC ===> __            SYSADOVC ===> __            SYSADEVC ===> __
     SYSAFCVC ===> __            SYSVECVC ===> __            SYSADJVC ===> __
     SYSADHVC ===> __            SYSADXVC ===> __            SYSADNVC ===> __
     SYSADMVC ===> __            SYSCPUVC ===> 8             SYSSRBVC ===> 10
     SYSTCBVC ===> 9_            SYSVELVC ===> __            SYSUGMVC ===> __
     SYSUGPVC ===> __            SYSUGDVC ===> __            SYSWGDVC ===> __
     SYSWGPVC ===> __            SYSDGMVC ===> __            SYSUJMVC ===> __
     SYSDJMVC ===> __            SYSDGEVC ===> __            SYSDGHVC ===> __
     SYSDGDVC ===> __            SYSDGJVC ===> __            SYSDGOVC ===> __
     SYSDGPVC ===> __            SYSDGSVC ===> __            SYSDGUVC ===> __
     SYSDGXVC ===> __            SYSDTLLN ===> __            SYSDTPSN ===> __
```

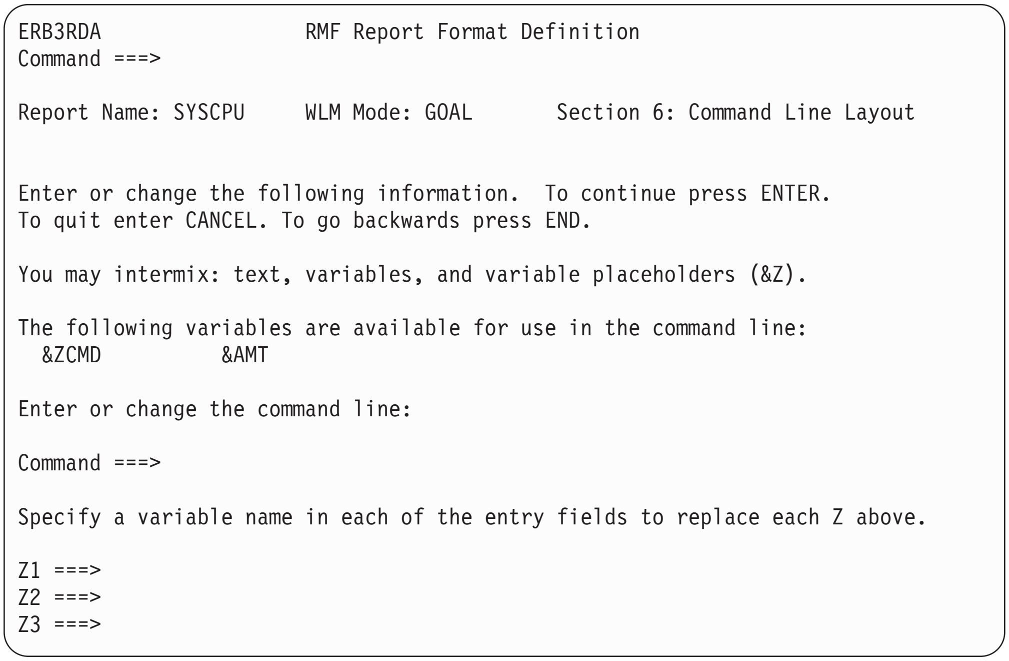
Figure 5-14. Report Column Layout Panel (ERB3RD9)

Command line layout panel (ERB3RDA)

Press ENTER to display the next panel ERB3RDA, the **Command Line Layout** panel.

On this panel, you can specify the format of the command line and scroll line as you want them to appear on the hardcopy of the tabular report. You must also define the command line and scroll line on the display panel of the tabular report.

Figure 5-15 is an example of Command Line Layout panel ERB3RDA.



```
ERB3RDA                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL       Section 6: Command Line Layout


Enter or change the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).

The following variables are available for use in the command line:
  &ZCMD          &AMT

Enter or change the command line:

Command ===>

Specify a variable name in each of the entry fields to replace each Z above.

Z1 ===>
Z2 ===>
Z3 ===>
```

Figure 5-15. Command Line Layout Panel (ERB3RDA)

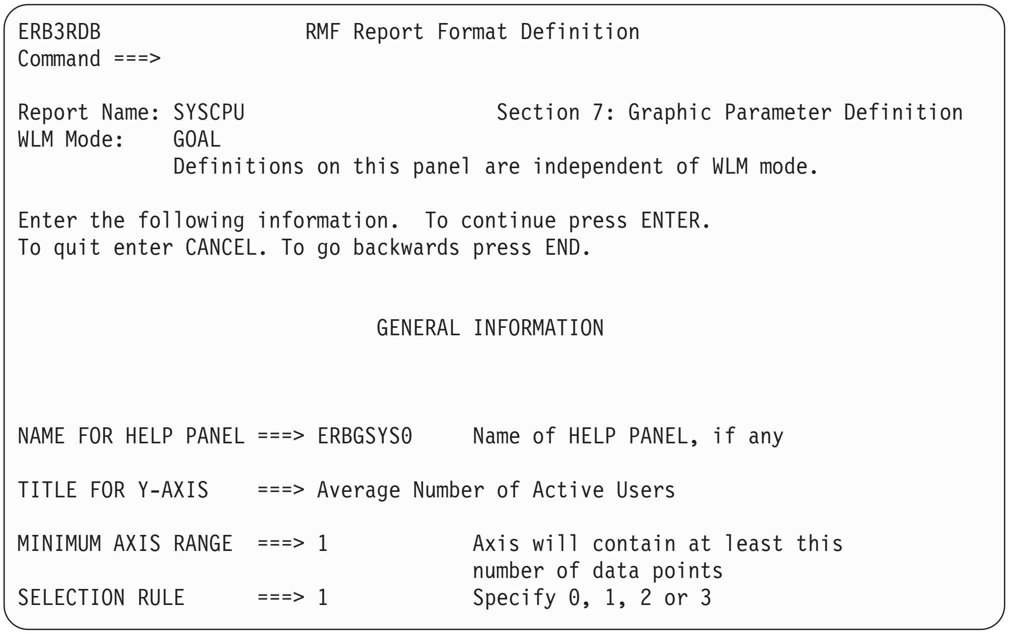
Graphic parameter definition panels (ERB3RDB, ERB3RDC, ERB3RDD)

If you specified BOTH or GRAPHIC for report mode on ERB3RD3, RMF displays the first **Graphic Parameter Definition** panel, ERB3RDB.

On this panel, you can specify general information about the graphic version of the report.

Note: If you specified TABULAR for report mode on the report format information panel (ERB3RD1) or used DI or WFEX as a prototype, the report format definition utility displays panel ERB3RDF. This panel allows you to save your changes and view the tabular report you have created or cancel your changes. See "Saving or cancelling changes on panel ERB3RDF" on page 5-28.

Figure 5-16 is an example of the Graphic Parameter Definition panel ERB3RDB that specifies general information for the graphic version of the SYSINFO report:



```
ERB3RDB                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                     Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.


                              GENERAL INFORMATION



NAME FOR HELP PANEL ===> ERBGSYS0     Name of HELP PANEL, if any

TITLE FOR Y-AXIS    ===> Average Number of Active Users

MINIMUM AXIS RANGE  ===> 1            Axis will contain at least this
                                      number of data points
SELECTION RULE      ===> 1            Specify 0, 1, 2 or 3
```

Figure 5-16. Graphic Parameter Definition Panel (ERB3RDB)

The fields and their meanings follow:

NAME FOR HELP PANEL

Specifies the name of the help panel that you provide for the graphic report. The field is optional.

TITLE FOR Y-AXIS

Specifies a line of text (maximum of 50 characters) to appear as a label for the bar graph in the graphic version of the report. Sample lines that appear in the graphic parameter table (ERBPTGS3) are:

- Percentage of Each User's Time
- Percentage of the User's Time
- Average Number of Active Users

For an example of the graphic parameter table (ERBPTGS3), see Chapter 7, "Monitor III data reporter tables", on page 7-1.

MINIMUM AXIS RANGE

Specifies the length of the bar graph depending on the text specified in TITLE FOR Y-AXIS as follows. For each line of text listed in the previous example, the minimum axis range is as follows:

- 100 for "Percent of Each User's Time"
- 100 for "Percent of the User's Time"
- 1 for "Average Number of Active Users"

If the length of the largest bar in the report exceeds the value you specify, RMF uses the length of the largest bar.

For an example of the graphic parameter table (ERBPTGS3), see Chapter 7, "Monitor III data reporter tables", on page 7-1.

SELECTION RULE

Specifies how the lines of the tabular report appear as bar graphs on the graphic version of the report. You can select one of the following values:

- 0 - One bar corresponds to one line of the RMF tabular report

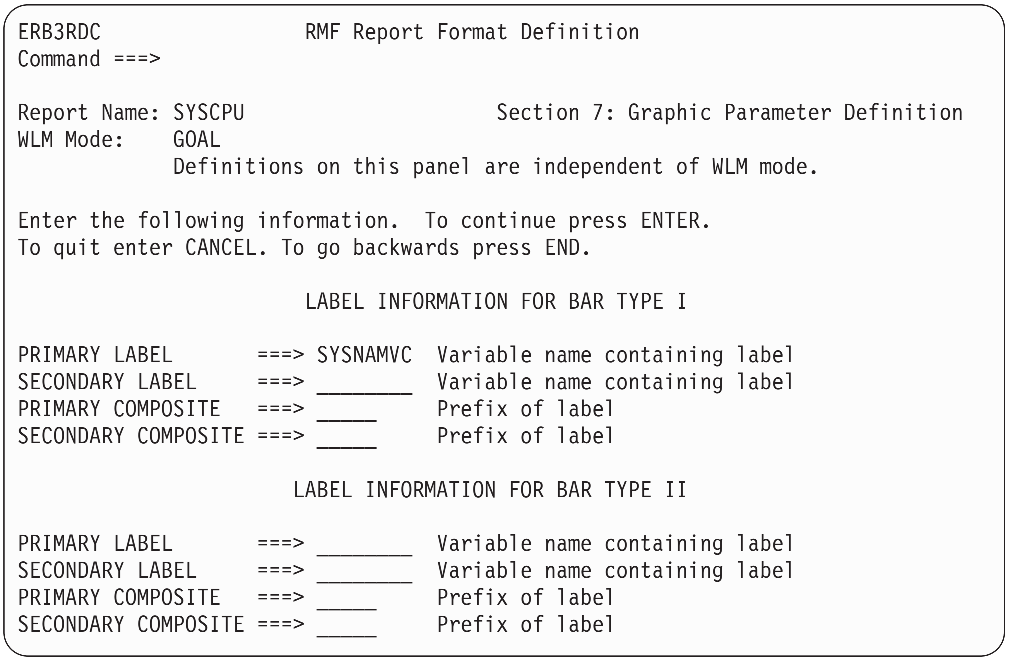
- 1 - One bar corresponds to one line of the RMF tabular report with sequence number 1 (for example, DEV, HSM, JES, STOR, PROC, DELAY, SYSINFO, and ENQ)
- 2 - One bar corresponds to the summary of logical lines of the report (for example, ENQR, DEVR reports)
- 3 - Two bar types can result from all logical lines of a logical block in the RMF tabular report (for example, STORR report) as follows:
  - Bar type 1 corresponds to a line of the tabular report with sequence number 1
  - Bar type 2 corresponds to each additional line of the logical block for a tabular report with a sequence number greater than 1

For an example of the graphic parameter table (ERBPTGS3), see Chapter 7, “Monitor III data reporter tables”, on page 7-1. For a description of logical line number and sequence number, see the panel field description for ERB3RD3 (Figure 5-8).

Press ENTER to display the next panel, ERB3RDC, the second **Graphic Parameter Definition** panel.

On this panel, you can specify labels for the graphic bars in the report. You can specify variable names for bar type 1 labels and bar type 2 labels.

Figure 5-17 is an example of the Graphic Parameter Definition panel ERB3RDC.



```
ERB3RDC                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                     Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

                        LABEL INFORMATION FOR BAR TYPE I

PRIMARY LABEL       ===> SYSNAMVC  Variable name containing label
SECONDARY LABEL     ===> ________  Variable name containing label
PRIMARY COMPOSITE   ===> _____     Prefix of label
SECONDARY COMPOSITE ===> _____     Prefix of label

                       LABEL INFORMATION FOR BAR TYPE II

PRIMARY LABEL       ===> ________  Variable name containing label
SECONDARY LABEL     ===> ________  Variable name containing label
PRIMARY COMPOSITE   ===> _____     Prefix of label
SECONDARY COMPOSITE ===> _____     Prefix of label
```

Figure 5-17. Graphic Parameter Definition Panel (ERB3RDC)

The panel fields and their meanings are as follows:

PRIMARY LABEL/SECONDARY LABEL
Specifies an 8 character variable name for a data value in the graphic version of the report. You can use the variable names that appear in the ISPF data table of the corresponding tabular report.

For example, in Figure 5-17, the Primary label will appear as average number of active users (SYNAMVC) on the graphic version of the SYSCPU report.

See Chapter 7, “Monitor III data reporter tables”, on page 7-1 for examples of the Graphic Parameter table (ERBPTGS3) and the RMF Report Data tables.

PRIMARY COMPOSITE/SECONDARY COMPOSITE

Specifies up to 5 characters of text as a prefix to the variable label specified in PRIMARY/SECONDARY LABEL. In Figure 5-17, no composite labels appear in the SYSCPU report. You can specify a prefix (like DMN for domain or PG for performance group) to appear in the graphic version of the report. The prefix is concatenated to the rightmost contents of the report table variable specified in PRIMARY/SECONDARY label.

See Chapter 7, “Monitor III data reporter tables”, on page 7-1 for examples of the Graphic Parameter table (ERBPTGS3) and the RMF Report Data tables.

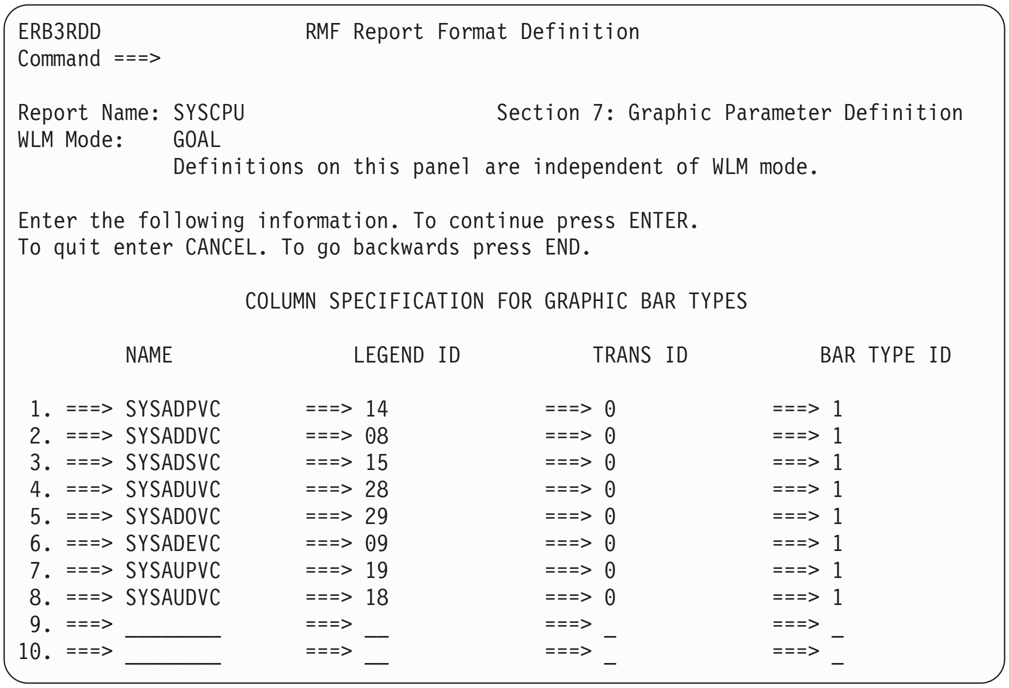
BAR TYPE refers to the number of bars used in the report depending on the logical line and sequence numbers.

See LOGICAL LINE/SEQUENCE NUMBER in the field descriptions for ERB3RD3 (Figure 5-8).

Press ENTER to display the next panel, ERB3RDD, the third **Graphic Parameter Definition** panel.

On this panel, you can specify data columns that you want to appear in the graphic version of the report.

Figure 5-18 is an example of the Graphic Parameter Definition panel ERB3RDD.



```
ERB3RDD                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                   Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information. To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

                   COLUMN SPECIFICATION FOR GRAPHIC BAR TYPES

         NAME              LEGEND ID           TRANS ID           BAR TYPE ID

 1. ===> SYSADPVC       ===> 14             ===> 0             ===> 1
 2. ===> SYSADDVC       ===> 08             ===> 0             ===> 1
 3. ===> SYSADSVC       ===> 15             ===> 0             ===> 1
 4. ===> SYSADUVC       ===> 28             ===> 0             ===> 1
 5. ===> SYSADOVC       ===> 29             ===> 0             ===> 1
 6. ===> SYSADEVC       ===> 09             ===> 0             ===> 1
 7. ===> SYSAUPVC       ===> 19             ===> 0             ===> 1
 8. ===> SYSAUDVC       ===> 18             ===> 0             ===> 1
 9. ===> ________       ===> __             ===> _             ===> _
10. ===> ________       ===> __             ===> _             ===> _
```

Figure 5-18. Graphic Parameter Definition Panel (ERB3RDD)

The panel fields and their meanings are as follows:

NAME

Specifies an 8 character variable name for a data value from the corresponding tabular report. This value will appear as a bar column in the graphic version of the report. The bar column can be a single bar (bar type 1) or a stacked bar (bar type 2) depending on what you specify for BARTYPE ID. See Chapter 7, "Monitor III data reporter tables", on page 7-1 for examples of RMF report data tables.

LEGEND ID
Specifies a number that corresponds to the color, pattern and the text of the graphic chart legend. Variables specified for NAME will appear in the color specified for LEGEND ID. You can specify a decimal value from 04 to 27; the numbers must match the color ID entries on the Color Graphic Option panels.

TRANS ID
Specifies a number that controls how the values for the variable in NAME are scaled on the bar graph in the graphic version of the report.

- 0 - value appears as is; no division is performed
- n - value is divided by 10^n where **n** equals an integer from 1 to 9.

See Chapter 7, "Monitor III data reporter tables", on page 7-1 for examples of RMF report data tables.

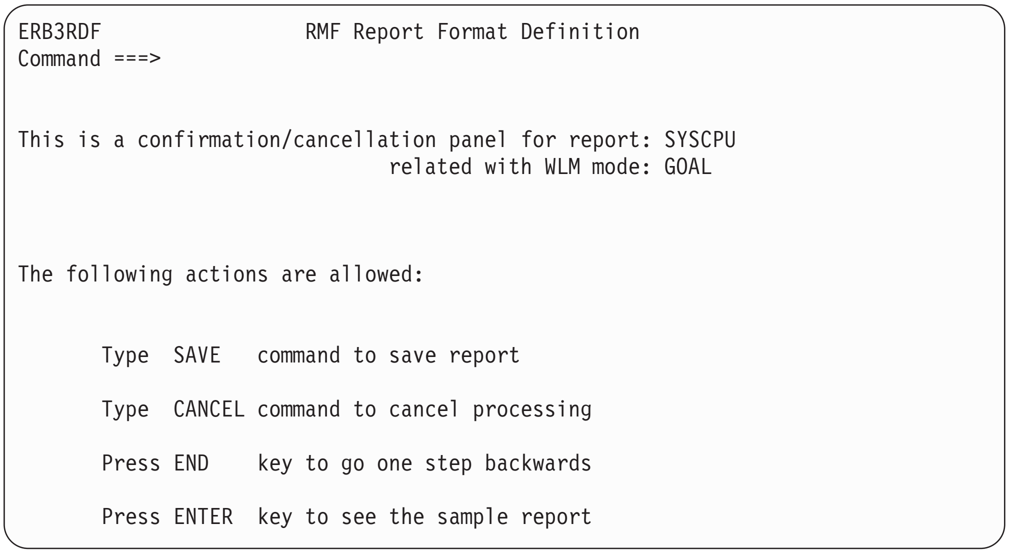
BARTYPE ID
Specifies a value that indicates where the data value for the variable in NAME appears for bar types in the graphic version of the report:

- 0 - indicates the value appears in both bar types
- 1 - indicates the value occurs in bar type 1
- 2 - indicates the value occurs in bar type 2

If you specified label information for only bar type 1 on the report parameter definition panel (ERB3RDC), you must specify bar type 1.

Saving or cancelling changes on panel ERB3RDF

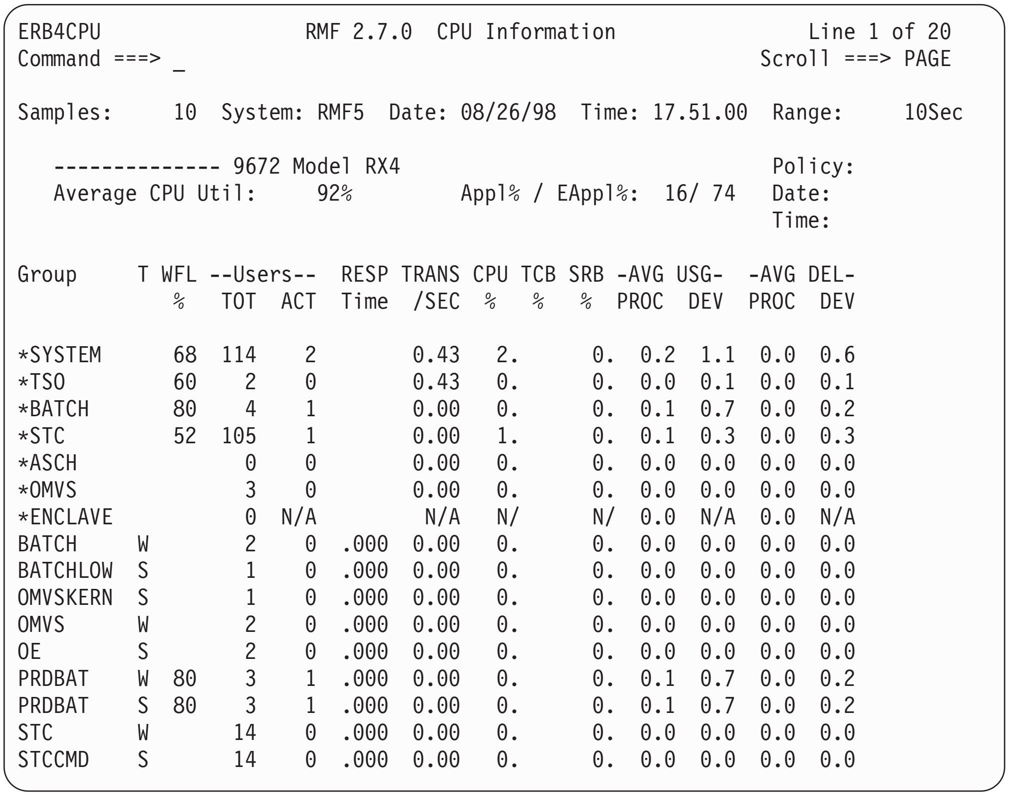
Once you have created or modified a report using the report format definition utility panels, RMF displays panel ERB3RDF, which allows you to confirm or cancel your changes.



```
ERB3RDF                RMF Report Format Definition
Command ===>


This is a confirmation/cancellation panel for report: SYSCPU
                               related with WLM mode: GOAL



The following actions are allowed:


       Type  SAVE   command to save report

       Type  CANCEL command to cancel processing

       Press END    key to go one step backwards

       Press ENTER  key to see the sample report
```

Figure 5-19. Configuration/Cancellation Panel (ERB3RDF)

You can get a report with sample data just to verify the correct layout of the report. In this example, some values are not displayed because they are not part of the sample data.



```
ERB4CPU                 RMF 2.7.0  CPU Information                Line 1 of 20
Command ===> _                                                Scroll ===> PAGE

Samples:     10  System: RMF5  Date: 08/26/98  Time: 17.51.00  Range:     10Sec

   -------------- 9672 Model RX4                               Policy:
   Average CPU Util:     92%        Appl% / EAppl%:  16/ 74   Date:
                                                               Time:

Group     T WFL --Users--  RESP TRANS CPU TCB SRB -AVG USG-  -AVG DEL-
             %   TOT  ACT  Time  /SEC  %   %   %  PROC  DEV  PROC  DEV

*SYSTEM      68  114    2        0.43   2.      0.  0.2  1.1  0.0  0.6
*TSO         60    2    0        0.43   0.      0.  0.0  0.1  0.0  0.1
*BATCH       80    4    1        0.00   0.      0.  0.1  0.7  0.0  0.2
*STC         52  105    1        0.00   1.      0.  0.1  0.3  0.0  0.3
*ASCH              0    0        0.00   0.      0.  0.0  0.0  0.0  0.0
*OMVS              3    0        0.00   0.      0.  0.0  0.0  0.0  0.0
*ENCLAVE           0  N/A         N/A   N/      N/  0.0  N/A  0.0  N/A
BATCH     W        2    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
BATCHLOW  S        1    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
OMVSKERN  S        1    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
OMVS      W        2    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
OE        S        2    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
PRDBAT    W  80    3    1  .000  0.00   0.      0.  0.1  0.7  0.0  0.2
PRDBAT    S  80    3    1  .000  0.00   0.      0.  0.1  0.7  0.0  0.2
STC       W       14    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
STCCMD    S       14    0  .000  0.00   0.      0.  0.0  0.0  0.0  0.0
```

Figure 5-20. Initial Version of the SYSCPU Report

The report shows that adjustments for some columns are necessary. You can do this either by stepping back to panel ERB3RD8 before you save the report or by modifying the stored report.

Enter SAVE to save the report or CANCEL to cancel your changes and return to the report definition initialization panel (ERB3RD1). If you save the report, RMF redisplays panel ERB3RD1 with a message that tells you the report has been modified or created. To exit the sample report panel and return to panel ERB3RDF, press END.

Deleting a user-defined report

If you specify DELETE for a report on the report format definition panel, RMF displays panel ERB3RDE. To confirm the deletion of the report, press ENTER and the report is deleted. To cancel the deletion, type CANCEL and press ENTER. RMF returns you to ERB3RD1.

Note: You can only delete a user-defined report. RMF does not allow you to delete an existing RMF report.

Ending the report utility

You can end the report format definition utility session by pressing END (PF3) on the report format definition panel (ERB3RD1) or by specifying CANCEL on any panel.

Implementing the report

To make the new SYSCPU report available, it needs to be integrated in a Monitor III selection panel. As defined initially, the report shall be added to the User Selection menu ERB3USR. You can do this by these modifications to the definition of the panel:

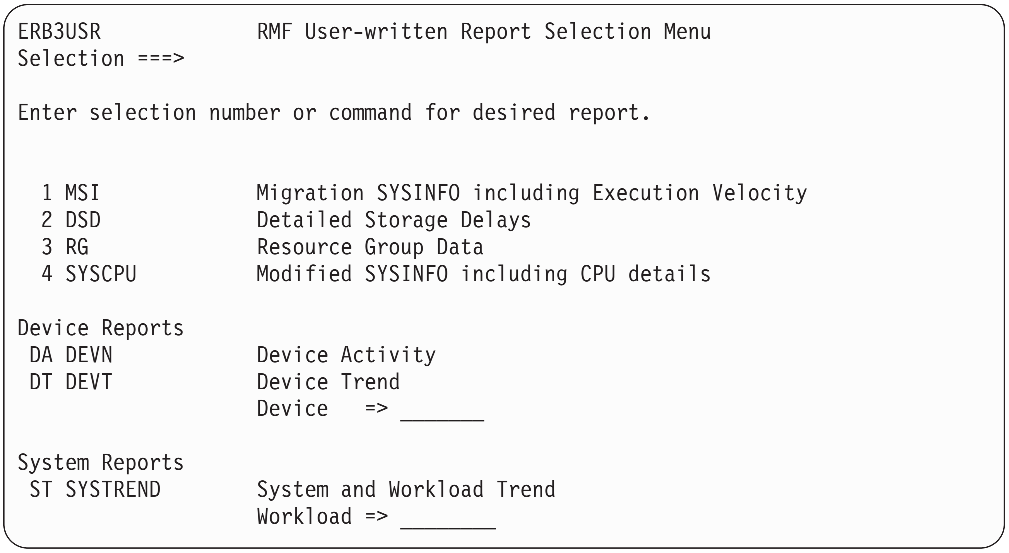
```
)attr default(!+_)
/***********************************************************************
/*          PANEL NAME: ERB3USR
/*

...

)body expand("") cmd(zcmd)
+          !         RMF User-written Report Selection Menu      " "
!Selection ===>_ZCMD " " +
+
<Enter selection number or command for desired report.
+
+
!  1<MSI           +Migration SYSINFO including Execution Velocity
!  2<DSD           +Detailed Storage Delays
!  3<RG            +Resource Group Data
!  4<SYSCPU        +Modified SYSINFO including CPU details
+

...
```

Figure 5-21. Modifications in User Selection Menu Definition (ERB3USR) - Part 1

```
 /* translate subsystem selections ********************************* */
    &erbcmdc = trans(&erbcmdc
                   1,'MSI'
                   2,'DSD'
                   3,'RG'
                   4,'SYSCPU'
                  ST,'SYSTREND'
                  DA,'DEVN'
                  DT,'DEVT'
                   *,*)
 /* Checks if command input is a valid RMF command.                  */
    ver(&erbcmdc,LIST, CANCEL, FIND, GRAPHIC, ICU, HARDCOPY, RESET,
        RFIND, TABULAR, TOGGLE,
        MSI, DSD, RG,  SYSCPU,
        DEVN, DEVT, SYSTREND,
        MSG=ERB562I)
 /* Checks if command input is a valid on this screen.               */
    ver(&erbcmdc,LIST,;
        MSI, DSD, RG,  SYSCPU,
        DEVN, DEVT, SYSTREND,
        MSG=ERB573I)

...

 /* selects action according to entered input ********************** */
    &zsel = trans(&erbcmdc
              MSI,'PGM(ERB3RDPC) PARM(MSI)'
              DSD,'PGM(ERB3RDPC) PARM(DSD)'
               RG,'PGM(ERB3RDPC) PARM(RG)'
           SYSCPU,'PGM(ERB3RDPC) PARM(SYSCPU)'
             DEVN,'PGM(ERB3RDPC) PARM(DEVN)'
             DEVT,'PGM(ERB3RDPC) PARM(DEVT)'
         SYSTREND,'PGM(ERB3RDPC) PARM(SYSTREND)'
                *,' ')
)END
```

Figure 5-22. Modifications in User Selection Menu Definition (ERB3USR) - Part 2

If you call the User Selection menu, you now get this new version:



```
ERB3USR             RMF User-written Report Selection Menu
Selection ===>

Enter selection number or command for desired report.


  1 MSI             Migration SYSINFO including Execution Velocity
  2 DSD             Detailed Storage Delays
  3 RG              Resource Group Data
  4 SYSCPU          Modified SYSINFO including CPU details

Device Reports
 DA DEVN            Device Activity
 DT DEVT            Device Trend
                    Device   => _______

System Reports
 ST SYSTREND        System and Workload Trend
                    Workload => ________
```

Figure 5-23. Modified User Selection Menu (ERB3USR)

Special considerations for modifying reports

If you want to add or delete lines in an existing RMF report or sort lines of a report without modifying the report heading, consider the following when you use the report format definition utility:

- Each RMF report data table (PHDRTAB1 in the phase driver table) contains the ISPF key type variables for the logical line number and line sequence number for the report. Each data table lists the logical lines and the sequence number(s) for logical lines of data in the report in ascending order. Sequence numbers for each logical line begin with 1. When you add, delete, or sort lines of an RMF report, be sure that the output table of your report (PHASE 1 or 2 TABLE on phase driver information panel ERB3RD2) arranges logical line and sequence numbers in ascending order.
- If you delete a line of a report with sequence number 1, you must also delete the logical line number of the report from the data table.
- If you want to rearrange the lines of an RMF report, you can use the ISPF service TBSORT as part of the CLIST you specify for phase 2. You can specify the CLIST with TBSORT on the phase driver information panel (ERB3RD2) as follows:

  ```
  CMD(mysort)
  ```

  where “mysort” is the name of your CLIST.

Installing your own phases

When you select a report during a reporter session, RMF uses ISPF SELECT services to generate report data tables and display the reports. You can supply your own routines for any of the 4 phases to produce user-defined reports. See “Data reporter phases” on page 5-8 for a description of the phases RMF invokes.

The following ISPF shared variables are available during all phases. They can be updated in Phase 1 by the Data Retrieval Service.

ERBDATE,ERBTIME
The ISPF shared pool variables that contain the beginning date and time of the display data.

ERBRANGE The ISPF shared pool variable that contains the range time of the display data. The beginning date/time plus the range time of the display data equals the ending date/time.

ERBSID The ISPF shared pool variable that contains the id of the system on which the data was collected.

ERBSAMPL The ISPF shared pool variable that contains the number of data samples for the time range.

ERBRMFVD The ISPF shared pool variable that contains the RMF version number of the data gatherer which collected the data. The format is **RMF VvRr** (i.e. RMF V1R2).

ERBDTBEG The ISPF shared pool variable which contains the beginning date/time value for the requested time range. The format is **MMDDYYYYHHMMSS** (i.e. 09252002183050 represents September 25, 2002 at 18:30:50).

ERBDTEND The ISPF shared pool variable which contains the ending date/time value for the requested time range. The format is **MMDDYYYYHHMMSS** (i.e. 08272002173010 represents August 27, 2002 at 17:30:10).

ERBMNTIM The ISPF shared pool variable which contains the Monitor III data gatherer MINTIME option value that was in effect when the data was gathered. The data is in external decimal format.

Phase 1

If you want to use your own program for phase 1, you must ensure that the ISPF shared pool variable PHDRPH1 contains the name of your program or CLIST. This variable appears in the phase driver table (ERBPHDS3) as an ISPF SELECT string. For RMF reports, the PARM value of the string matches the name of the RMF report command. You can use the report definition format utility to specify your own PHASE 1 SELECT STRING. See Chapter 7, “Monitor III data reporter tables”, on page 7-1 for an example of the phase driver table (ERBPHDS3) entries and how they are specified.

If you want to change the time range from which your data is collected, you can invoke the Data Retrieval Service (ERB3RDRS) module from your phase 1 program. See “Data retrieval service (ERB3RDRS)” on page 5-36 for information about how to invoke the Data Retrieval Service.

The following ISPF shared variables contain information that RMF uses to generate a report during phase 1:

ERBREPC The ISPF shared pool variable that contains the current command or report selection. RMF uses this variable as a key to ERBPHDT3, the phase driver table. This table has an entry (in the table field PHREPNA) for each RMF command or report selection. RMF retrieves the necessary information to generate the report during phase 1 from ERBPHDT3 (a copy of ERBPHDS3).

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

ERBSSHG3 The ISPF shared pool variable that contains the address of the set-of-samples header (SSHG3). This control block contains the addresses of the sample data that correspond to the time and range values specified during the Monitor III data gatherer session or as indicated by the Data Retrieval Service. (See Figure 5-2 for an example of ERBSSHG3.)

ERBSUPP The ISPF shared pool variable that contains the number of the subpool that non RMF functions must use for GETMAINS.

During phase 1, the phase driver module (ERB3RPH1) uses the information in the report row entry of ISPF table ERBPHDT3 (a copy of ERBPHDS3) to produce the RMF report. If you supplied your own program or CLIST for phase 1, that routine gets control.

Upon completion, phase 1 must provide the following output:

ERBREPC The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET1 The ISPF shared pool variable that should contain your return code from the program or CLIST used in phase 1.

For RMF supplied reports, ERB3RPH1 creates the report in phase 1 and returns one of the following return codes:

0 ISPF table successfully generated for the report

4 ISPF table generated for the report has some data, but errors have occurred

8 ISPF table generated for the report has no data, and an error has occurred

For your own routine, you might want to use the same return codes.

PHDRTAB1 The ISPF shared pool variable that contains the name of the ISPF data table generated in phase 1. If you omit phase 2, you must ensure that PHDRTAB2 contains the same name as PHDRTAB1. See phases 2 and 3 described later.

You can define your own ISPF shared pool variables to contain information that you want to include for phase 1. To ensure that no interference with RMF created variables occurs, the first three characters of user-defined variables should appear as follows:

```
USR
```

Phase 2

For phase 2, you supply a program or CLIST to modify the ISPF table created for the report in phase 1.

The following ISPF shared variables contain information for phase 2:

ERBREPC The ISPF shared pool variable that should contain the current command or report selection.

ERBCMDC,ERBPARC
The ISPF shared pool variables that should contain the current command (ERBCMDC) and any command parameters (ERBPARC).

PHDRET1 The ISPF shared pool variable that should contain your return code from the program or CLIST used in phase 1.

For RMF supplied reports, ERB3RPH1 creates the report in phase 1 and returns one of the following return codes:

0 ISPF table successfully generated for the report

4 ISPF table generated for the report has some data, but errors have occurred

8 ISPF table generated for the report has no data, and an error has occurred

For your own routine, you might want to use the same return codes.

PHDRTAB1 The ISPF shared pool variable that contains the name of the ISPF data table generated in phase 1.

Upon completion, phase 2 must provide the following output:

ERBREPC The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET2 The ISPF shared pool variable that should contain the return codes from the RMF program or CLIST used to create the report in phase 2.

PHDRTAB2 The ISPF shared pool variable that should contain the name of the ISPF data table generated in phase 2. You can use the same table name as for PHDRTAB1.

Phase 3

For phase 3, RMF formats the ISPF table generated in phase 1 or 2 and displays the report. To format the ISPF report data tables, RMF uses the tabular report format table (ERBFMTS3), the RMF header table (ERBHDRS3), and the graphic parameter table (ERBPTGS3). The RMF display phase module (ERB3RDSP) displays the report by means of the ISPF TBDISPL service.

The following ISPF shared variables contain information that you can use to format and display a report during phase 3:

ERBREPC
The ISPF shared pool variable that contains the current command or report selection. The variable is a key to obtain formatting information for the tabular report in the report format table (ERBFMTS3) or the graphic report in the graphic parameter table (ERBPTGS3). For examples of these tables, see Appendix B.

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

PHDRET1,PHDRET2
The ISPF shared pool variables that should contain return codes from phase 1 and 2.

PHDRTAB2
The ISPF shared pool variable that should contain the name of the ISPF data table generated in phase 1 and/or phase 2.

SESRPFU3
The ISPF shared pool variable that contains the report mode (TABULAR or GRAPHIC).

RMF uses module ERB3RDSP to display the reports. The module dynamically constructs a panel from information in the format tables. It builds header and model lines and constructs the graphic area within the panel and uses the ISPF data table whose name appears in the ISPF shared pool variable PHDRTAB2.

Upon completion, phase 3 must provide the following output:

ERBREPC
The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET3
The ISPF shared pool variable that should contain the return code from the program or CLIST used to format and display the report.

If you decide to replace the RMF module ERB3RDSP with your own routine, you must consider the following:

- To obtain a display of your reports in GO mode, you must invoke the ISPF service CONTROL LOCK before the ISPF service TBDISPL is performed.
- Your module must handle all ISPF PASSTHRU commands.

Phase 4

For phase 4, you provide a program that can perform cleanup services for resources you might have used during previous phases. For example, if you have used ISPF TBCREATE with the WRITE SHARE option to create an ISPF table, you can use ISPF TBEND to delete the table during phase 4. Or use TBEND to save and then delete the table. See the ISPF publications that describe these services for more information.

The following ISPF shared variables contain information that you can use to format and display a report during phase 4:

ERBREPC The ISPF shared pool variable that contains the current command or report selection.

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

ERBSUPP The ISPF shared pool variable that contains the number of the subpool used for GETMAINS.

PHDRET1,PHDRET2,PHDRET3
The ISPF shared pool variables that should contain return codes from phase 1, 2, and 3.

Upon completion, phase 4 must provide the following output:

ERBREPC The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET4 The ISPF shared pool variables that should contain return codes from phase 4.

Data retrieval service (ERB3RDRS)

The Data Retrieval Service (ERB3RDRS) module provides flexibility for user exits to change the time range from which data is collected. The module is called from phase 1 of your user exit. This service can be invoked by either calling it,

Example

```
CALL ERB3RDRS (PARMAREA)
```

or by using the ISPF SELECT service.

Example

```
ISPEXEC SELECT PGM(ERB3RDRS) PARM(PARMAREA)
```

To use this service, the caller must invoke the module ERB3RDRS with the registers and parameter area described in “Parameter area contents” on page 5-37.

Programming considerations

Do not link the module ERB3RDRS to your application program. Assembler programs must use LOAD or LINK macros to access the module; PL/I programs must use FETCH/RELEASE; and C programs must use the builtin function FETCH.

The caller must be in 31-bit addressing mode and can run unauthorized.

Function codes

The function code specifies the time range to be used by the Data Retrieval Service:

1 Most recent number of MINTIMEs (as in GO mode)

2 Retrieve data from the range determined by BEG Date and Time and END Date and Time (similar to the BREF command with parameters DATE=,TIME=, and RANGE=)

3 Retrieve data from the range determined by using END Date and Time as end time, and going backward in time using the current RANGE (similar to the BREF command without parameters)

4 Retrieve data from the range determined by BEG Date and Time as begin time, and going forward in time using the current RANGE (similar to the FREF command without parameters)

Registers at entry

The contents of the registers on entry to this service are:

Register	Contents
0	Not used
1	Parameter list address
2-12	Not used
13	Standard save area address
14	Return address
15	Entry point address of ERB3RDRS

Parameter area contents

The parameter area passed by the caller to the RMF Data Retrieval Service is a single character string, preceded by a halfword containing the length of the parameter area in binary. The parameter area is as follows:

First word Bytes 0 to 3: function code

Second word Bytes 4 to 7: number of MINTIMEs (this is used only with function code 1)

Character string
Bytes 8 to 21: begin date and time of the requested time range in character format of MMDDYYYYHHMMSS.

Character string
Bytes 22 to 35: end date and time of the requested time range in character format of MMDDYYYYHHMMSS.

Output

The Data Retrieval Service module updates the following shared pool variables:

ERBSSHG3 The ISPF shared pool variable that contains the beginning address of the common set of samples. If no data could be retrieved, this variable is set to hexadecimal zero.

ERBDTBEG The ISPF shared pool variable that contains the beginning date/time value of the retrieved range.

ERBDTEND The ISPF shared pool variable that contains the ending date/time value of the retrieved range.

ERBMNTIM The ISPF shared pool variable that contains the Monitor III data gatherer MINTIME option value in external format.

Return codes

Upon return from this service, register 15 provides the return code and reason code as listed in Table 5-1:

- Bytes 0 and 1 are not used (x‘0000’)
- Byte 2 contains reason code
- Byte 3 contains return code

Retrieval service

Table 5-1. Return and Reason Codes for the Data Retrieval Service (ERB3RDRS)

Return Code (Decimal)	Reason Code (Decimal)	Description
0	0	Data returned with no errors.
4	4	Data might be inconsistent due to a SET IPS change detected within the specified range. This is valid for data being gathered with RMF Version 4.
8		Data only partially returned.
	8	Partial data returned. Message ERB589I displayed.
	9	VSAM retrieval error occurred. Partial data returned. Message ERB589I displayed.
	13	The WLM service policy has changed, or the IPS values have been modified. This is valid for data being gathered with RMF Version 5 and above.
	14	The RMF cycle time has changed.
	15	IPL detected.
12		No data returned.
	8	No data returned. Message ERB587I displayed.
	9	VSAM retrieval error occurred. No data returned. Message ERB587I displayed.
	14	Cycle time changed. Message ERB559I displayed.
	15	IPL detected. Message ERB558I displayed.
	16	No data available. Message ERB591I displayed
	17	Data gatherer is not active. Message ERB565I displayed.
	18	Preallocated data sets are unusable. Message ERB583I displayed.
	19	Preallocated data sets found to be unusable during data retrieval. Message ERB583I displayed.
	20	Too many reporters tried to access the in-storage buffer. Message ERB564I displayed.
	21	Retrieval from in-storage buffer failed. Message ERB564I displayed.
	22	No data is in the in-storage buffer. Message ERB591I displayed.
	23	Insufficient storage to copy data from the in-storage buffer. Message ERB564I displayed.
16	0	Incorrect function code.

Note: The RMF Monitor III standard reports provide information on the same time range as was requested in the last use of the Data Retrieval Service.

End of Programming Interface information

TSO/E user authorization

Programming Interface information

TSO/E must be installed on your system to use the following commands.

All the data collected and reported by RMF during a Monitor III display session is obtained from commonly addressable storage that is not fetch protected. However, if your installation wants to limit the use of the command that starts an RMF Monitor III session under TSO/E, one method available is to replace the RMF control section with your own module. For Monitor III you replace the control section ERB3SOCK. Your routine will then be invoked as part of the RMF response to the RMF command.

ERB3SOCK (Monitor III) runs in problem state with a key of 8. When this control section gets control, register 1 points to a two-word address list. The first address points to the seven-byte user ID of the user who has entered the RMF command. The second word points to the PSCB. Figure 5-24 illustrates the input parameter structure.

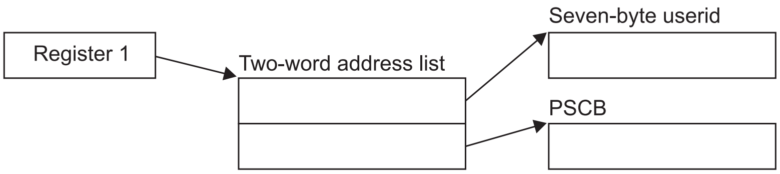


Figure 5-24. ERB3SOCK Input Parameter Structure

The module that you code to replace ERB3SOCK must be reenterable. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.

The processing your module performs depends on the method you choose to validate the user. Possible methods include issuing a RACHECK, prompting the user for a password, or checking the userid against a list of valid userids. Information on the TSO/E services available to perform these functions, such as TGET or TPUT, can be found in *z/OS TSO/E Programming Services*.

You can also use the PSCB bits defined for user use. This field (PSCBATR2 in the PSCB) comes from the UADS and can be updated by the USERDATA keyword of the ADD and CHANGE subcommands of the ACCOUNT command. See *z/OS TSO/E System Programming Command Reference* for more information on these commands.

When your routine has completed its processing, set a return code of 0 in register 15 to indicate to RMF that the user is authorized to enter RMF. Set a return code of 4 in register 15 to indicate to RMF that the user is not authorized to enter RMF. In response to this return code, RMF will display a message at the display station. No session will be started. After setting the appropriate return code, return control by branching on the contents of register 14.

For the Monitor III TSO/E session the user authorization exit routine (ERB3SOCK) is part of the RMF load module that contains the RMF command. This module resides in SYS1.SERBLINK as load module RMF; its entry point is ERB3RTSO. Before your authorization routine can run, you must link edit it with RMF; the control statements required are:

```
(ERB3SOCK object deck)
INCLUDE ddname(RMF)
ENTRY ERB3RTSO
NAME RMF(R)
```

End of Programming Interface information

Chapter 6. Using Monitor III VSAM data set support

About VSAM Data Sets

This chapter:

- Describes the data set structure and content for the Monitor III data set support function
- Lists in table form the record fields and table entries associated with data set support

See the *z/OS RMF User's Guide* for more information about data set support and recording.

Data set record structure

Programming Interface information

If no specific limitation is stated, then all fields in the records, including those indicated as RESERVED FOR USER, but **excluding** all others indicated as RESERVED are part of the programming interface.

With the data set support function, RMF uses VSAM relative record data sets (RRDS) to record measurement information during a Monitor III gatherer session.

During data set recording RMF collects measurement data in the form of one set of samples for each MINTIME and records the samples on the VSAM data sets. Before storing the data, RMF compresses the data one MINTIME at a time. The data is stored in compressed format except for the Data Set Header and Index Table (ERBDSIG3) and the MINTIME Set of Samples Header Table (ERBSSHG3). The description of the data tables are valid only after the RMF decompression interface (ERB3RDEC) is used to decompress the data one MINTIME at time. The RMF Monitor III reporter will decompress the data after retrieving it from the VSAM data sets. To directly access the VSAM data sets and process them without the use of the Monitor III reporter, use the service module, ERB3RDEC. See "Data set decompression" on page 6-3 for more information.

RMF data can be accessed directly by relative record number or by sequential records. Each data set is a string of fixed-length records, and each record is identified by a relative record number. Because RMF treats the data it records on the data set as a linear data set, it writes the logical records as a contiguous stream of sampled data with little dependency on the record size. To allow retrieval of the data, an index relates the time stamp of every MINTIME set of samples with the offset of the set of samples within the data set and its length; therefore, you can determine the relative record number of any given set of samples within a data set by dividing the offset and the length of the set of samples by the record length, which is 32,752 bytes. (Note: VSAM does not maintain the index.)

The first record on every VSAM data set contains the data set header. It is followed by the index information (see "ERBDSIG3 - Data set header and index" on page 6-22). RMF builds one index entry for each MINTIME set of samples in the data set. When no more entries can fit into the index, RMF closes the data set. The records in the data set following the index information contain the measurements of each MINTIME set of samples (see "ERBSSHG3 - MINTIME set of samples header" on page 6-51). RMF stores data on the data set as follows:

- contiguously arranges MINTIME sets of samples in chronological order
- stores the data so that one MINTIME may cross record boundaries

Figure 6-1 shows an example of how these records can be arranged on a Monitor III VSAM data set.

```
 Header
 and Index|  MINTIME 1 | MINTIME 2 | MINTIME n  |
|*********|************|***********|....********|
|         |         |         |         |         |        |
|Record 1 |Record 2 |Record 3 |Record 4 |  .....  |Record n|
```

Figure 6-1. Monitor III Data Set Record

Record processing requires reading the header (record 1) and index to obtain the offset and length of a selected MINTIME set of samples. The record(s) containing the MINTIME sets of samples must be read into contiguous storage before RMF can process them. MINTIME 2 starts in record 3 and ends in record 4. Note that before MINTIME processing can begin, both records 3 and 4 must be read into contiguous storage.

End of Programming Interface information

Data set decompression

Programming Interface information

The MINTIME set-of-samples stored on VSAM data sets is compressed by RMF prior to storing the data. For direct access of the VSAM data sets and processing without use of the Monitor III reporter, you will need to use the Data Set Decompression Interface Service module, ERB3RDEC.

To use this service, the caller must invoke the module ERB3RDEC with the registers and parameter area described in “Parameter area contents”. The service returns only *one* record to the caller, which contains all the data.

Programming considerations

Do not link the module ERB3RDEC to your application program. Assembler programs must use LOAD or LINK macros to access the module; PL/I programs must use FETCH/RELEASE; and C programs must use the built-in function FETCH.

The caller must be in 31-bit addressing mode and can run unauthorized.

Registers at entry

The contents of the registers on entry to this service are:

Register	Contents
0	Reserved
1	Parameter list address
2-12	Reserved
13	Standard save area address
14	Return address
15	Entry point address of ERB3RDEC

Parameter area contents

The parameter area passed by the caller to the RMF Data Set Decompression Interface Service is a 3-fullword string, preceded by a halfword containing the length of the parameter area. The parameter area is as follows:

First word	Bytes 0 to 3: address of the compressed set-of-samples
Second word	Bytes 4 to 7: address of output area for decompressed set-of-samples
Third word	Bytes 8 to 11: length of output area

Output

ERB3RDEC returns the following information in the parameter area depending on the return code (RC):

Third word

RC=0: length of the output area for the decompressed set-of-samples.

RC=4: minimum length required for the output area to hold the decompressed set-of samples.

RC>4: the bytes remain unchanged.

Return codes

Upon return from this service, register 15 provides return codes listed in Table 6-1.

Table 6-1. Return Codes for the Data Set Decompression Interface Service

Return Code (Decimal)	Description
0	Decompression successful, length of decompressed set-of-samples returned.
4	Decompression unsuccessful. The output area was too small to hold the decompressed set-of-samples. The minimum length required to hold the decompressed set-of-samples is returned. Obtain a larger output area and try again.
8	Decompression unsuccessful. Address passed for the compressed set-of-samples points to an uncompressed set-of-samples.
12	Decompression unsuccessful. Address passed for the compressed set-of-samples does not point to a valid set-of-samples.

Coded example

The following Assembler code example calls the Data Set Decompression Interface Service twice. The first call obtains the required length of the output area for the specified decompressed set-of samples. The second call performs the decompression.

This sample code assumes that register 2 points to the address of the compressed set-of-samples. It can be included in your installation's data retrieval code.

```
* Assuming, register 2 points to the compressed set-of-samples
         MVC      INRECA,0(R2)         Pointer to input record
* Calls Decompress Routine to retrieve the length of the
* uncompressed record.
         LA       R1,OUTAREA           Address of uncompressed record
         ST       R1,OUTRECA           Stores address in parmlist
         MVC      OUTRECL,INITLNG      Length of uncompressed record
         LA       R1,PARMADDR          Parameter to R1
         LINK     EP=ERB3RDEC          Invokes decompress routine
* Checks Return Code
         ST       R15,RETCODE          Saves return code
         CLC      R15,=F'4'            Checks return code
         BNE      PROCESS              Output area NOT too small
* Allocates required output area
         L        R3,OUTRECL           Required output length
         SR       R4,R4                Subpool 0
         GETMAIN  RU,LV=(3),SP=(4)     Get storage
         ST       R1,OUTRECA           Address of uncompressed record
* Calls Decompress Routine
         LA       R1,PARMADDR          Parameter to R1
         LINK     EP=ERB3RDEC          Invokes decompress routine
```

```
* Checks Return Code
         ST       R15,RETCODE          Saves return code
         LTR      R15,R15              Tests return code
         BZ       PROCESS              Decompress successful
* Decompress not successful. Releases output area
         L        R2,OUTRECA           Area address
         L        R3,OUTRECL           Area length
         SR       R4,R4                Subpool 0
         FREEMAIN RU,LV=(3),A=(2),,SP=(4)
PROCESS  DS       0H
* Check return code and process the decompressed record here.
* OUTRECA contains the address of the uncompressed record and the
* return code from ERB3RDEC is in RETCODE.
                  ...
* Declarations for the coding example above
INITLNG  DC       F'100'               Initial length
OUTAREA  DS       CL100                Initial output area
PARMADDR DC       A(PARMLIST)          Address of parameter list
RETCODE  DS       F                    Return code
         CNOP     2,4                  Alignment
PARMLIST DC       H'12'                Length of parameter area. This
*                                      field has to be initialized
*                                      with the decimal value 12.
INRECA   DS       F                    First word. It has to be
*                                      initialized with the address of
*                                      the compressed set-of-samples.
OUTRECA  DS       F                    Second word. It has to be
*                                      initialized with the address of
*                                      the output area which holds the
*                                      uncompressed set-of-samples.
OUTRECL  DS       F                    Third word. It has to be
*                                      initialized with the size of
*                                      the output area. ERB3RDEC will
*                                      return the size of the un-
*                                      compressed set-of-samples in
*                                      this field.

* Registers
R0       EQU      0
R1       EQU      1
R2       EQU      2
R3       EQU      3
R4       EQU      4
R5       EQU      5
R6       EQU      6
R7       EQU      7
R8       EQU      8
R9       EQU      9
R10      EQU      10
R11      EQU      11
R12      EQU      12
R13      EQU      13
R14      EQU      14
R15      EQU      15
```

End of Programming Interface information

Data set content

Programming Interface information

A MINTIME set of samples collected during the Monitor III gatherer session can be formatted and displayed during a Monitor III reporter display session. Each MINTIME set of samples is independent of other MINTIME sets of samples, and if you specify the same MINTIME value as that of the RANGE period for a display session, the report displays the information for that MINTIME set of samples collected during the gatherer session. Measurement values for each MINTIME set of samples are organized as tables or records, the formats of which appear at the end of this chapter. The types of measurement tables or records are:

ERBASIG3	ASID table
ERBCATG3	Cache data information table
ERBCFIG3	Coupling facility information table
ERBCPUG3	Processor data control block
ERBCSRG3	Common storage remaining table
ERBDSIG3	Data set header and index
ERBDVTG3	Device table
ERBENCG3	Enclave data table
ERBENTG3	Enqueue name table
ERBGEIG3	General information table
ERBGGDG3	Global gatherer data table
ERBPGPER	Performance group period table
ERBREDG3	Resource data record
ERBSHDG3	Sample header
ERBSSHG3	MINTIME set of samples header
ERBUWDG3	USE/WAIT record
ERBXMHG3	Moved samples header control block

Each is described in "Monitor III data set record and table formats" on page 6-9. Each offset is from the beginning of the table that contains the offset. Clock times are local from the time-of-day (TOD) clock.

Figure 6-2 shows the relationships between the Monitor III data set support tables and records.

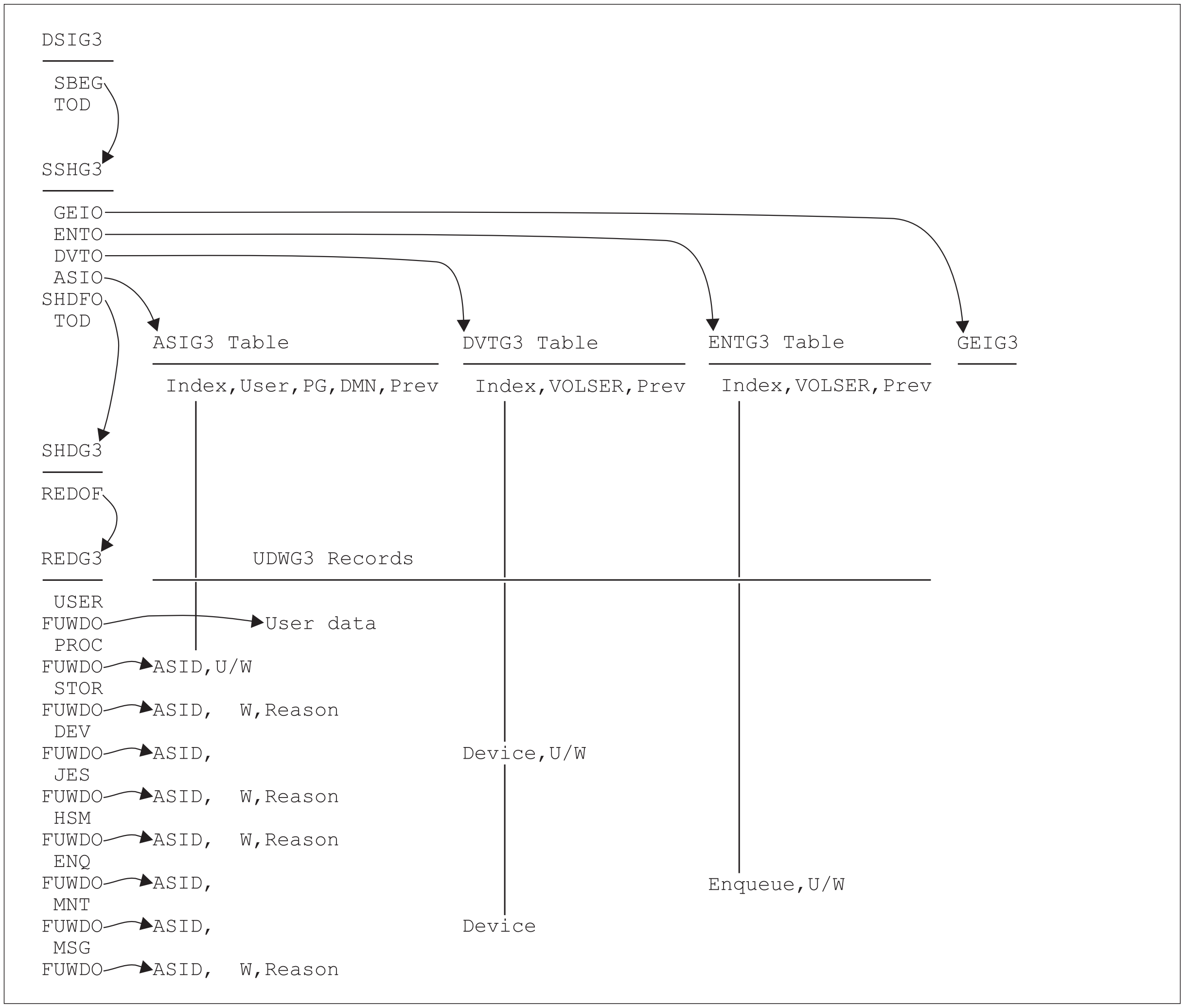


Figure 6-2. Monitor III Measurement Table and Record Relationships

The data set header and index (ERBDSIG3) describe the available measurement times (MINTIME sets of samples) and the data set offsets of each MINTIME set of samples header (ERBSSHG3).

The MINTIME set of samples header (ERBSSHG3) contains offsets to the address space id table (ERBASIG3), the device table (ERBDVTG3), enqueue name table (ERBENTG3), the general information table (ERBGEIG3), a group of sample headers (ERBSHDG3), the common storage remaining table (ERBCSRG3), and the performance group period table (PGPER). These tables describe information about each MINTIME interval within a data set.

Each sample header (ERBSHDG3) describes one sample CYCLE, and sample headers (ERBSHDG3) within one MINTIME are chained together by offsets.

The resource records (ERBREDG3) contain information about sampling for each resource. RMF first samples each type of hardware and software resource; RMF then samples user-written exit routines. The sample header (ERBSHDG3) for user-written exit routines contains an offset to the first resource record.

VSAM data

RMF creates in sequence one USE/WAIT record (ERBUWDG3) for each entry it finds in the queue for each resource. The resource record (ERBREDG3) contains an offset to the first USE/WAIT record for each resource.

The address space id table (ERBASIG3) contains one entry for each ASID/job/performance group (PG)/domain (DMN) combination. Each table entry contains the ASID number, its own index, and the index of the previous table entry for the ASID. (During one MINTIME interval, a job could exit, then reenter the system and therefore be assigned the same ASID. In this case, the job could have two sets of table entries for that MINTIME.)

The device table (ERBDVTG3) contains an entry for each device/VOLSER combination. Each entry contains the device number, its own index, and the index of the previous table entry for the device.

RMF correlates USE/WAIT records with their current table entries also by index.

To obtain the offset of each entry within the ASIG3 or DVTG3 table, multiply the length of each table entry by the index (see Figure 6-2).

```
Index x length of table entry.
```

For higher level languages, ASIG3 or DVTG3 arrays can be accessed with the index and an origin of 0.

To obtain the offset of each entry within the ENTG3 table, multiply the length of each table entry by the index (see Figure 6-2) minus 1:

```
(Index - 1) x length of table entry.
```

For higher level languages, the ENTG3 array can be accessed with the index and an origin of 1.

The common storage remaining table (ERBCSRG3) contains one entry for each job that ended and did not release all common storage. Each table entry contains the ASID number, the jobname, the JES-ID, the termination date, the termination time, and the amount of remaining common storage.

The performance group period table (ERBPGPER) contains performance group period related information gathered from SRM’s workload activity measurement table (WAMT). Each table entry contains the performance group number, the period number, the domain number, the elapsed time for all transactions that ended in the performance group period, the time spent on JES or APPC queues by all transactions that ended in the performance group period, and the number of transactions that ended in the performance group period.

End of Programming Interface information

Monitor III data set record and table formats

Programming Interface information

This section describes the measurement records and tables used for the Monitor III data set support function. Fields that are reserved for RMF are used for debugging purposes, for maintaining the data areas, or do not contain RMF Monitor III report data.

Note: The following record and table mappings apply only to the current release and are subject to change for future releases.

ERBASIG3 - Address space identification table

Offsets		Name	Length	Format	Description
Dec	Hex				
ASIG3 Header Section:					
0	0	ASIASIG3	5	EBCDIC	Acronym ‘ASIG3’
5	5	ASIVERG3	1	binary	Control block version X'0E'
6	6	ASIHDRLE	1	binary	Length of ASIG3 header
7	7	*	1	*	Reserved
8	8	ASIENTMX	4	binary	Number of table entries
12	C	ASIENTNR	4	binary	Index of last table entry
16	10	ASIENTLN	4	binary	Length of one entry
20	14	ASISSTVO	4	binary	Offset to service-class-served table
24	18	*	8	EBCDIC	Reserved
32	20	ASIENTRY(*)	328	EBCDIC	Array of all ASID table entries
ASIG3 Table Entry Section:					
0	0	ASIENIDX	2	binary	Index of this table entry
2	2	ASIPREVI	2	binary	Index of the previous table entry for the same address space (ASID)
4	4	ASIJOBNA	8	EBCDIC	Jobname for this address space id (ASID). This and the next 5 offsets describe the sort criteria for the address space (ASID). RMF creates a new entry whenever the JOBNAME, PG (performance group), or DMN (domain) changes for the address space.
12	C	ASINPG	2	binary	Control performance group
14	E		1		Reserved
15	F	ASIDMNN	1	binary	Domain
16	10	ASIASINR	2	binary	ASID number

ERBASIG3 - ASID table

Offsets		Name	Length	Format	Description
Dec	Hex				
18	12	ASIFLAG1	2	binary	Job flags **Bit** **Meaning When Set** **0** Started task **1** Batch job **2** TSO ASID **3** ASCH ASID **4** OMVS ASID **5-15** Reserved
20	14	ASICPUTA	4	binary	Total TCB+SRB time (in milliseconds)[1]
24	18	ASIDCTIA	4	binary	Total channel connect time (in 128 microsecond units)[1]
28	1C	ASIFIXA_VE	4	floating point	Number of central fixed frames[1]
32	20	ASITRCA	4	binary	Total number of transactions[1]
36	24	ASIFMCT_VE	4	floating point	Number of frames for swapped-in users[1]
40	28	ASIFMCTI_VE	4	floating point	Number of frames for idle users[1]
44	2C	ASIESF_VE	4	floating point	Number of expanded storage frames for swapped-in users[1]
48	30	ASIESFI_VE	4	floating point	Number of expanded storage frames for idle users[1]
52	34	ASISMPCT	2	binary	Number of valid samples
54	36	ASISWAP	2	binary	Number of samples when job was physically swapped-out
56	38	ASIIDLE	2	binary	Number of samples when job was idle
58	3A	ASISWAR	2	binary	Number of samples when job was swapped-out ready
60	3C	ASIACT	2	binary	Active using or delayed count
62	3E	ASIUKN	2	binary	Number of samples when job status was unknown
64	40	ASISUSEN	2	binary	Number of single state using samples
66	42	ASISUCPR	2	binary	Number of single state samples using processor (PROC)
68	44	ASISUCDV	2	binary	Number of single state samples using device (DEV)
70	46	ASISWAIN	2	binary	Number of single state samples delayed by any resource
72	48	ASISDCPR	2	binary	Number of single state samples delayed by the processor (PROC)
74	4A	ASISDCDV	2	binary	Number of single state samples delayed by device (DEV)
76	4C	ASISDCST	2	binary	Number of single state samples delayed by paging or swapping (STOR)
78	4E	ASISDCJE	2	binary	Number of single state samples delayed by JES
80	50	ASISDCHS	2	binary	Number of single state samples delayed by HSM

Offsets		Name	Length	Format	Description
Dec	Hex				
82	52	ASISDCEN	2	binary	Number of single state samples delayed by ENQ
84	54	ASIVECTA	4	binary	Total accumulated vector processor time
88	58	ASISDCSU	2	binary	Number of single state samples delayed by SUBS
90	5A	ASISDCOP	2	binary	Number of single state samples delayed by OPER
92	5C	ASISDCMS	2	binary	Number of single state samples delayed by OPER MESSAGE
94	5E	ASISDCMT	2	binary	Number of single state samples delayed by OPER MOUNT
96	60	ASIPAGES	2	binary	Page delay
98	62	ASISWAPS	2	binary	Swap delay
100	64	ASIDIV_VE	4	floating point	Number of DIV frames
104	68	ASIAUXSC_VE	4	floating point	Number of auxiliary slots
108	6C	ASIPINA	4	binary	Page-in counts
112	70	ASIDIVCT	2	binary	Number of DIV invocations
114	72	ASIACTHF	2	binary	Number of address spaces active and holding storage counter
116	74	ASISWAPI	2	binary	Number of address spaces swapped in (not logically and not physically swapped
118	76	ASISDCXC	2	binary	Single state delayed by XCF - part of subs
120	78	ASIJCLAS	8	EBCDIC	Job class, Source: OUCBCLS
128	80	ASIPINES	4	binary	Expanded storage page-in count
132	84	ASIFLAG2	4	binary	Common storage flags **Bit** **Meaning When Set** **0** CSA amounts incomplete **1** SQA amounts incomplete **2** APPC initiator **3** BATCH initiator **4-31** Reserved
136	88	ASICSASC	4	binary	CSA sample count
140	8C	ASISQASC	4	binary	SQA sample count
144	90	ASICSAA	4	floating point	CSA allocation
148	94	ASISQAA	4	floating point	SQA allocation
152	98	ASIECSAA	4	floating point	ECSA allocation
156	9C	ASIESQAA	4	floating point	ESQA allocation
160	A0	ASIJLCYC	4	binary	Time-offset when this job was last found in the system, expressed in CYCLE time units.
164	A4	ASIJOBST	8	EBCDIC	Job selection time in GMT
172	AC	ASIJESID	8	EBCDIC	JES ID
180	B4	ASITET	4	binary	Transaction elapsed time, in 1024 microsecs units
184	B8	ASISRBTA	4	binary	Total accumulated SRB time

ERBASIG3 - ASID table

Offsets		Name	Length	Format	Description
Dec	Hex				
188	BC	ASIIOCNT	4	binary	IO count
192	C0	ASILSCT	2	binary	Count of ″long″ logical swaps
194	C2	ASIESCT	2	binary	Count of ″long″ swaps to expanded storage
196	C4	ASIPSCT	2	binary	Count of ″long″ physical swaps
198	C6	ASILSCF	4	floating point	Sum of all central frames for logically swapped user at all samples.
202	CA	ASILSEF	4	floating point	Sum of all expanded frames for logically swapped user at all samples.
206	CE	ASILSSA	2	binary	Total logically swapped samples
208	D0	ASIPSEF	4	floating point	Sum of all expanded frames for swapped user (except logical) at all samples.
212	D4	ASIPSSA	2	binary	Total swapped samples (except logical)
214	D6	ASIORTI	2	binary	STOR/OUTR delay samples for swap reason 1: Terminal input wait
216	D8	ASIORTO	2	binary	STOR/OUTR delay samples for swap reason 2: Terminal output wait
218	DA	ASIORLW	2	binary	STOR/OUTR delay samples for swap reason 3: Long wait
220	DC	ASIORXS	2	binary	STOR/OUTR delay samples for swap reason 4: Aux. storage shortage
222	DE	ASIORRS	2	binary	STOR/OUTR delay samples for swap reason 5: Real storage shortage
224	E0	ASIORDW	2	binary	STOR/OUTR delay samples for swap reason 6: Detected long wait
226	E2	ASIORRQ	2	binary	STOR/OUTR delay samples for swap reason 7: Requested swap
228	E4	ASIORNQ	2	binary	STOR/OUTR delay samples for swap reason 8: Enqueue exchange swap
230	E6	ASIOREX	2	binary	STOR/OUTR delay samples for swap reason 9: Exchange swap
232	E8	ASIORUS	2	binary	STOR/OUTR delay samples for swap reason 10: Uniliteral swap
234	EA	ASIORTS	2	binary	STOR/OUTR delay samples for swap reason 11: Transition swap
236	EC	ASIORIC	2	binary	STOR/OUTR delay samples for swap reason 12: Improve central storage usage
238	EE	ASIORIP	2	binary	STOR/OUTR delay samples for swap reason 13: Improve system paging rate
240	F0	ASIORMR	2	binary	STOR/OUTR delay samples for swap reason 14: Make room for an out too long user
242	F2	ASIORAW	2	binary	STOR/OUTR delay samples for swap reason 15: APPC wait
244	F4	ASIORIW	2	binary	STOR/OUTR delay samples for swap reason 16: OMVS input

Offsets		Name	Length	Format	Description
Dec	Hex				
246	F6	ASIOROW	2	binary	STOR/OUTR delay samples for swap reason 17: OMVS output
248	F8	ASIRCLX	2	binary	Report-class-list index
250	FA	ASIORSR	2	binary	STOR/OUTR delay samples for swap reason 18: In-real swap
252	FC	ASICPUC	2	binary	CPU capping delay
254	FE	ASIACOM	2	binary	Common paging
256	100	ASIAPRV	2	binary	Private paging
258	102	ASIAVIO	2	binary	VIO paging
260	104	ASIASWA	2	binary	Swapping
262	106	ASIUNKN	2	binary	Unknown count for calculating execution velocity
264	108	ASICCAP	2	binary	Resource capping delay
266	10A	ASICQUI	2	binary	Quiesce delay
268	10C	ASIAXM	2	binary	Cross-memory delay
270	10E	ASIAHSP	2	binary	Hiperspace delay
272	110	ASICUSE	4	binary	CPU using
276	114	ASITOTD	4	binary	Total delays for calculating execution velocity
280	118	ASISRVO	4	binary	Offset from service-class-served table-header to corresponding row
284	11C	ASITOTSV	4	floating point	Total number of shared page views in this address space
288	120	ASISVINR	4	floating point	Total number of shared pages in central storage that are valid for this address space
292	124	ASISPVLC	4	floating point	Total number of shared page validations in this address space
296	128	ASIGSPPI	4	floating point	Total number of shared page-ins from auxiliary storage for this address space
300	12C	ASIGASPD	2	binary	Single state samples delayed for shared storage paging
302	12E	*	2	*	Reserved
304	130	ASIOREPL	4	binary	Number of outstanding replies
308	134	ASITOTU	4	binary	Number of multi-state using samples
312	138	ASIIOU	4	binary	Number of multi-state I/O using samples
316	13C	ASIASSTA	4	binary	Additional SRB time
320	140	ASIPHTMA	4	binary	Preemptable-class SRB time

ERBASIG3 - ASID table

Offsets		Name	Length	Format	Description
Dec	Hex				
324	144	ASIMSTS	4	binary	Miscellaneous states. **Bit** **Meaning When Set** **0** Address space is OMVS related **1** Address space matched a classification rule in the active policy which prevents managing the region based on the response time goals of its served transactions **2** CPU protection was assigned either to the address space or to transaction service classes being served by the space, and SRM is honoring the protection **3** Storage protection was assigned either to the address space or to transaction service classes being served by the space, and SRM is honoring the protection **4** This address space provides service to transactions classified to a different class than the address space itself **5** WLM is managing this address space to meet the goals of work in other service classes **6-31** Reserved

[1] Sum of all values obtained at each sample. To obtain average values, divide by the number of valid samples (ASISMPCT).

ERBCATG3 - Cache data information table

Offsets		Name	Length	Format	Description
Dec	Hex				
CAT Table (CATG3) Header Section:					
0	0	CATG3_Acro	5	EBCDIC	Acronym 'CATG3'
5	5	CATG3_Ver	1	binary	Control block version X'01'
6	6	*	2	EBCDIC	Reserved
8	8	CATG3_Tot_Len	4	binary	Total length including this header, the maximum number of CATG3 array entries and all SMF 74 subtype 5 records
12	C	*	4	binary	Reserved
16	10	CATG3_Hdr_Len	2	binary	Length of this header
18	12	CATG3_Entry_Len	2	binary	Length of one CATG3 array entry
20	14	CATG3_Entry_Num	4	binary	Number of CATG3 array entries in use
24	18	CATG3_Max_Num	4	binary	Maximum number of CATG3 array entries
28	1C	*	20	EBCDIC	Reserved
48	30	CATG3_LDTO	8	binary	Offset GMT to local time (STCK format)
56	38	*	8	EBCDIC	Reserved
CAT Table (CATG3) Array Entry:					
0	0	CATG3_SSID	2	binary	Cache subsystem ID (SSID)
2	2	CATG3_MDL	1	binary	Subsystem model
3	3	*	1	EBCDIC	Reserved
4	4	CATG3_B_TOD	8	binary	Cache interval GMT start time (STCK format)[1]
12	C	CATG3_E_TOD	8	binary	Cache interval GMT end time (STCK format)[1]
20	14	CATG3_Msg_Flg	2	binary	If Bit 8 is set, no SMF 74.5 record data available for this SSID
22	16	CATG3_Stat_Code	2	binary	Status code. 0 = SMF 74.5 record data available 4 = IOS return code given 8 = IDCSS01 return code given 98 = System or User Abend given
24	18	CATG3_RC_IOS	2	binary	IOS return code
26	1A	CATG3_RC_IDCSS	2	binary	IDCSS01 return code
28	1C	*	1	EBCDIC	Reserved
29	1D	CATG3_CMPC	3	binary	Bit 0-11: System completion code Bit 12-23: User completion code
32	20	CATG3_Rec_VF	4	binary	Offset to SMF 74 subtype 5 record with Cache Subsystem Activity data
36	24	CATG3_Rec_Lng	4	binary	Length of SMF 74 subtype 5 record

[1] Device reserve activity can cause a data gatherer interface to wait until a RESERVE has been released. This in turn can cause the cache interval to be longer than the set of samples interval (see SSHTIBEG and SSHTIEND). To convert this value to local time, add CATG3_LDTO (see header section).

ERBCFIG3 - Coupling facility information table

Offsets		Name	Length	Format	Description
Dec	Hex				
CFI Table (CFIG3) Header Section:					
0	0	CFICFIG3	5	EBCDIC	Acronym 'CFIG3'
5	5	CFIVERG3	1	binary	Control block version X'03'
6	6	CFIHDRLE	2	binary	Length of CFIG3 header section
8	8	CFITOTLE	4	binary	Lenght of total CFIG3
12	C	CFIENTLE	2	binary	Length of one CFI entry
14	E	CFIENTNR	2	binary	Number of CFI entries
16	10	CFISTROF	4	binary	Offset to first structure entry CFISTRUS
20	14	CFISTRLE	2	binary	Length of one structure entry
22	16	CFISTRNR	2	binary	Number of structure entries
24	18	CFISTEOF	4	binary	Offset to first structure extension entry CFISTRES. If extensions exist, there is one CFISTRES entry for a CFISTRUS entry.
28	1C	CFISTELE	2	binary	Length of one structure entry
30	1E	CFISTENR	2	binary	Number of structure entries
32	20	CFICONOF	4	binary	Offset to first structure connection entry CFICONNS
36	24	CFICONLE	2	binary	Length of one CFICONNS entry
38	26	CFICONNR	2	binary	Number of CFICONNS entries
40	28	*	24	EBCDIC	Reserved
CFI Table (CFIG3) Entry Section:					
0	0	CFIENNAM	8	EBCDIC	Name of CF
8	8	CFIENSYS	8	EBCDIC	Name of system (from SYSNAME parameter of IEASYSxx member)
16	10	CFIENST1	4	binary	Index of first CFISTRUS/CFISTRES entries of this CF
20	14	CFIENST#	4	binary	Number of CFISTRUS/CFISTRES
24	18	CFIENLVL	4	binary	CFLEVEL of microcode
28	1C	CFIENMOD	4	EBCDIC	CF model
32	20	CFIENVER	3	EBCDIC	CF version
35	23	CFIENPAM	1	binary	Path used mask for CFIENPID

Offsets		Name	Length	Format	Description
Dec	Hex				
36	24	CFIENFLG	2	binary	Status flags **Bit** **Meaning When Set** **0** Coupling Facility was connected to the system at the end of the mintime **1** Coupling Facility became active during the mintime **2** Coupling Facility structure hardware data gathered by RMF master gatherer on this member **3** Coupling Facility structure hardware data gathered with CFDETAIL option **4** Coupling Facility structure hardware data snapshot values taken from SMF records **5** CHPIDs set offline during the interval - CFIENPBC contains an invalid value **6-15** Reserved
38	26	CFIENSCG	2	binary	Number of subchannels defined in the I/O gen
40	28	CFIENSCU	2	binary	Number of subchannels currently in use
42	2A	CFIENSCL	2	binary	Number of subchannels that can be used (limit)
44	2C	CFIENPID(8)	1	binary	CF links. Valid entries have corresponding bit in CFIENPAM set
52	34	CFIENPNR	4	binary	Number of online processors
56	38	CFIENBSY	4	binary	Busy time in milliseconds
60	3C	CFIENWAI	4	binary	Wait time in milliseconds
64	40	CFIENPBC	4	binary	Number of times CF requests failed due to path busy
68	44	CFIENSCC	4	binary	Subchannel contention count (all subchannel busy)
72	48	CFIENTSD	4	binary	Total amount of CF storage defined (units = 4K byte blocks)
76	4C	CFIENTSF	4	binary	Amount of free CF storage (units = 4K byte blocks)
80	50	CFIENTOR	4	binary	Number of total requests
84	54	CFIENTID(8)	1	binary	CF link type. Valid entries have corresponding bit in CFIENPAM set
CFISTRUS Table Entry Section:					
0	0	CFISTNAM	16	EBCDIC	Name of connected structure in this CF
16	10	CFISTVER	8	binary	Structure Version number

ERBCFIG3 - Coupling facility data

Offsets		Name	Length	Format	Description
Dec	Hex				
24	18	CFISTTYP	1	binary	Structure type: 1 = unserialized list 2 = serialized list 3 = lock 4 = cache 5 = unknown
25	19	*	5	EBCDIC	Reserved
28	1C	CFISTFLG	1	binary	Status Flags **Bit** **Meaning When Set** **0** Structure was connected to the system at the end of the mintime **1** Structure became active during the mintime **2-7** Reserved
29	1D	*	1	EBCDIC	Reserved
30	1E	CFISTEIN	2	binary	Index of according CFI table entry
32	20	CFISTTRC	4	binary	Total number of user requests completed
36	24	CFISTARC	4	binary	Count of number of times for async. requests executed by CF
40	28	CFISTATM	8	binary	Summed service time for asynchronous requests in microseconds
48	30	CFISTSRC	4	binary	Count of number of times for sync. requests executed by CF
52	34	CFISTSTM	8	binary	Summed service time for synchronous requests in microseconds
60	3C	CFISTSTA	4	binary	Number of requests changed from synchronous to asynchronous
64	40	CFISTQRC	4	binary	Count of number of times for queued requests
68	44	CFISTDRC	4	binary	Number of times a request was found delayed in case of dump serialization
72	48	CFISTCN	4	binary	Lock structure only: Number of times any request encountered lock contention
76	4C	CFISTFCN	4	binary	Lock structure only: Number of times any request encountered false lock contention (storage contention within the structure)
CFISTRES Table Entry Section:					
0	0	CFIENCN1	4	binary	Index of first CFICONNS entry belonging to this structure
4	4	CFIENCN#	4	binary	Number of CFICONNS entries belonging to this structure
8	8	CFISTSIZ	4	binary	Allocated size of structure (units = 4K byte blocks)
12	C	CFISTMAE	4	binary	List structure only: Maximum number of elements
16	10	CFISTCUE	4	binary	List structure only: Current number of elements in use

Offsets		Name	Length	Format	Description
Dec	Hex				
20	14	CFISTLEL	4	binary	List + Lock structure only: Limit on number of list entries
24	18	CFISTLEM	4	binary	List + Lock structure only: Maximum number of list entries used during mintime
28	1C	CFISTLTL	4	binary	Lock + serialized List structure only Limit on number of lock table entries
32	20	CFISTLTM	4	binary	Lock + serialized List structure only Maximum number of lock table entries used during mintime
36	24	CFISTDEN	4	binary	Cache structure only: Total directory entry count
40	28	CFISTDEL	4	binary	Cache structure only: Total data element count
44	2C	CFISCDEC	4	binary	Cache structure only: Current directory entry count
48	30	CFISCDAC	4	binary	Cache structure only: Current data element count
52	34	CFISCRHC	4	binary	Cache Read hit Counter
56	38	CFISCWHC	4	binary	Cache Write hit counter
60	3C	CFISCDER	4	binary	Cache Directory entry reclaim counter
64	40	CFISCXIS	4	binary	Cache XI counter
68	44	CFISCCOC	4	binary	Cache Castout Counter
CFICONNS Table Entry Section:					
0	0	CFICNSIN	2	binary	Index of according CFISTRUS table entry
2	2	*	2	EBCIDC	Reserved
4	4	CFICNSYS	8	EBCIDC	Name of connecting system
12	C	CFICNNAM	16	EBCDIC	Name of connection
28	1C	CFICNJOB	8	EBCDIC	Name of connecting job
36	24	CFICNASI	2	binary	ASID of connecting job
38	26	CFICNSTA	2	binary	Connection status: 0 = Not known 1 = Active 2 = Failed Persistent 3 = Failing 4 = Disconnecting
40	28	CFICNCFL	4	binary	CF Level requested for connect

ERBCPUG3 - Processor data control block

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	CPUG3_AC	5	EBCDIC	Name of CPUG3
5	5	CPUG3_VE	1	binary	Control block version X'04'
6	6	*	2	*	Reserved
8	8	CPUG3_HDRL	4	binary	Header length
12	C	CPUG3_TOTL	4	binary	Total length this area
16	10	CPUG3_NUMPRC	8	binary	Number of processors (online during total mintime) multiplied by mintime (in microseconds)
24	18	CPUG3_LOGITI	8	binary	Logical CPU time in microseconds. This is the sum of MVS NON_WAIT time of all online logical processors in the time range
32	20	CPUG3_PHYSTI	8	binary	Physical CPU time in microseconds. This is the sum of all CPU times used by all logical processors. In the case of a native (non PR/SM) system this time is equal to the logical CPU time
40	28	CPUG3_STATUS	4	binary	Status information **Bit** **Meaning When Set** **0** BASIC mode system **1** LPAR mode system **2** Running as VM guest **3** Gatherer had permanent error **4** Diagnose 204 failed **5** VARY activity seen during the range. The number of logical processors used to accumulate the CPU time values varied. **6** Diagnose 204 extended format available **7** No MSU data available **8** Does not contain CPCDB data area **9-31** Reserved
44	2C	CPUG3_PRCON	4	binary	Number of online processors at end of mintime
48	30	CPUG3_NUMPRCOL	4	binary	Accumulated number of online processors. To get average number, divide by number of samples
52	34	CPUG3_NUMVECOL	4	binary	Accumulated number of online vector processors. To get average number, divide by number of samples
56	38	*	72	*	Reserved

ERBCSRG3 - Common storage remaining table

Offsets		Name	Length	Format	Description
Dec	Hex				
CSRG3 Header Section:					
0	0	CSRCSRG3	5	EBCDIC	Acronym ‘CSRG3’
5	5	CSRVERG3	1	binary	Control block version X'01'
6	6	*	1	*	Reserved
8	8	CSRHDRLE	2	binary	Length of CSRG3 header
10	A	CSRENTLE	2	binary	Length of one entry
12	C	*	4	*	Reserved
16	10	CSRENTNR	4	binary	Index of last available entry
20	14	*	12	*	Reserved
CSRG3 Table Entry Section:					
0	0	CSRASINR	2	binary	ASID number
2	2	*	2	*	Reserved
4	4	CSRJOBNA	8	EBCDIC	Jobname
12	C	CSRJESID	8	EBCDIC	JES-ID, taken from JSAB
20	14	CSRTDATE	4	EBCDIC	Ending Date, packed decimal OYYYYDDD, see documentation of the ’TIME’ macro
24	18	CSRTTIME	4	EBCDIC	Ending Date, packed decimal HHMMSSth, see documentation of the ’TIME’ macro
28	1C	CSRCSA	4	binary	CSA amount
32	20	CSRSQA	4	binary	SQA amount
36	24	CSRECSA	4	binary	ECSA amount
40	28	CSRESQA	4	binary	ESQA amount
44	2C	CSRFLAG	2	binary	Common Storage Flags **Bit** **Meaning** **0** CSA amounts complete **1** SQA amounts complete **2-15** Reserved
46	2E	*	2	*	Reserved

ERBDSIG3 - Data set header and index

Offsets		Name	Length	Format	Description
Dec	Hex				
Data Set Header Section					
0	0	DSIDSIG3	5	EBCDIC	Acronym ‘DSIG3’
5	5	DSIGRMFV	1	binary	Control block version X'02'
6	6	DSIGID	4	EBCDIC	System identifier
10	A	*	2	*	Reserved
12	C	DSIGTODC	8	binary	Time data set was created
20	14	DSIGTODF	8	binary	Time stamp for first set of samples
28	1C	DSIGTODL	8	binary	Time stamp for last set of samples
36	24	DSIGFSPT	4	binary	Offset of first set of samples from ERBDSIG3
40	28	DSIGLSPT	4	binary	Offset of last set of samples from ERBDSIG3
44	2C	DSIGNEPT	4	binary	Offset of next set of samples to be written
48	30	DSIGFIPT	4	binary	Offset of the first index entry from ERBDSIG3
52	34	DSIGLIPT	4	binary	Offset of the last index entry from ERBDSIG3
56	38	DSIGNIPT	4	binary	Offset of next index to be written
60	3C	DSIGILEN	4	binary	Length of an index entry
64	40	DSIGINUS	4	signed	Number of current index to set of samples
68	44	DSIGTDSF	8	EBCDIC	Time stamp of first policy
76	4C	DSIGTDSL	8	EBCDIC	Time stamp of last policy
84	54	DSIGFPPT	4	signed	Offset to start of first policy
88	58	DSIGLPPT	4	signed	Offset to start of the last policy
92	5c	DSIGFPIP	4	signed	Offset to first policy index
96	60	DSIGLPIP	4	signed	Offset to last policy index
100	64	DSIGNPIP	4	signed	Offset to next policy index
104	68	DSIGCIPN	4	signed	Current index number to policy
108	6C	DSIGFIPN	4	signed	First index number to policy
112	70	DSIGSPLX	8	EBCDIC	Sysplex-ID of this system
120	78	DSIGSPXD	32	EBCDIC	Reserved for sysplex
152	98	*	104	*	Reserved
Data Set Index Section					
0	0	DSIGTOD1	8	EBCDIC	Time stamp for start of set of samples or service policy
8	8	DSIGTOD2	8	EBCDIC	Time stamp for end of set of samples or service policy
16	10	DSIGSBEG	4	binary	Offset from the start of the data set to the start of the set of samples or start of the service policy

Offsets		Name	Length	Format	Description
Dec	Hex				
20	14	DSIGSLEN	4	binary	Physical (possibly compressed) length of the set of samples or length of service policy as contained in SVPDLE
24	18	DSIGFLG	1	binary	Data set flags **Bit** **Meaning** **0** Service policy index **1-7** Reserved
25	19	*	3	*	Reserved

ERBDVTG3 - Device table

Offsets		Name	Length	Format	Description
Dec	Hex				
Device Table Header Section:					
0	0	DVTDVTG3	5	EBCDIC	Acronym ‘DVTG3’
5	5	DVTVERG3	1	binary	Control block version X'08'
6	6	DVTHDRLE	1	binary	Length of the device table (DVTG3) header
7	7	DVTENTLE	1	binary	Length of each table entry
8	8	DVTENTMX	4	binary	Number of table entries
12	C	DVTENTNR	4	binary	Index of last table entry
16	10	DVTENTRY	104	EBCDIC	Entry in the device table
Device Table (DVTG3) Entry Section:					
0	0	DVTVOLI	6	EBCDIC	VOLSER for this device
6	6	DVTENIDX	2	binary	Index of this table entry
8	8	DVTDEVNR	2	binary	Device number in hexadecimal format
10	A	DVTPREVI	2	binary	Index of the previous table entry for the same device
12	C	DVTSMPCT	4	binary	Number of valid samples
16	10	DVTSMPNR	4	binary	Sample sequence number
20	14	DVTFLAG1	1	binary	Device type indicator **Bit** **Meaning When Set** **0** Reserved **1** DASD device **2** TAPE device **3** Number of alias exposures for a PAV device has changed **4** Virtual DASD **5** Reserved **6** LCU number is valid **7** Multiple exposure device (PAV)
21	15	DVTFLAG2	1	binary	Device storage indicators — these flags indicate if the time values in offsets 24 through 60 are available. **Bit** **Meaning When Set** **0** CONN/DISC/PEND time values at begin time available **1** CONN/DISC/PEND time values at end time available **2** DEV BUSY DELAY/CUB DELAY/DPB DELAY time values at end time available **3** DEV BUSY DELAY/CUB DELAY/DPB DELAY time values at end time available **4** Device has PLPA page data sets **5** Device has COMMON page data sets **6** Device has LOCAL page data sets **7** Reserved
22	16	DVTMEXNR	2	binary	Number of base and alias volumes

Offsets		Name	Length	Format	Description
Dec	Hex				
24	18	DVTDISIF	4	binary	Device DISC time at begin of the MINTIME for this set of samples (in 128-microsecond units)
28	1C	DVTPETIF	4	binary	Device PEND time at begin of the MINTIME for this set of samples (in 128-microsecond units)
32	20	DVTCOTIF	4	binary	Device CONN time at begin of the MINTIME for this set of samples (in 128-microsecond units)
36	24	DVTDVBIF	4	binary	Device busy delay time at begin of the MINTIME for this set of samples (in 128-microsecond units)
40	28	DVTCUBIF	4	*	No longer used
44	2C	DVTDISIL	4	binary	Device DISC time at end of the MINTME for this set of samples (in 128-microsecond units)
48	30	DVTPETIL	4	binary	Device PEND time at end of the MINTIME for this set of samples (in 128-microsecond units)
52	34	DVTCOTIL	4	binary	Device CONN time at end of the MINTIME for this set of samples (in 128-microsecond units)
56	38	DVTDVBIL	4	binary	Device busy delay time at end of the MINTIME for this set of samples (in 128-microsecond units)
60	3C	DVTCUBIL	4	*	No longer used
64	40	DVTTYP	4	EBCDIC	Device type mapped by the UCBTYP macro
68	44	DVTIDEN	8	EBCDIC	Device identification (device model)
76	4C	DVTCUID	8	EBCDIC	Control unit model
84	54	DVTSPBIF	4	*	No longer used
88	58	DVTSPBIL	4	*	No longer used
92	5C	DVTIOQLC	4	binary	I/O queue length count
96	60	DVTSAMPA	4	binary	Accumulated I/O instruction count
100	64	*	2	*	Reserved
102	66	DVTLCUNR	2	binary	LCU number
104	68	DVTSAMPP	4	binary	I/O instruction count (previous value)
108	6C	DVTCMRIF	4	binary	Initial command response time first
112	70	DVTCMRIL	4	binary	Initial command response time last

ERBENCG3 - Enclave data table

Offsets		Name	Length	Format	Description
Dec	Hex				
ENCARRAY					
0	0	ENCG3ACR	5	EBCDIC	Acronym 'ENCG3'
5	5	ECCG3VER	1	binary	Control block version X'03'
6	6	*	2	*	Reserved
8	8	ENCG3TLN	4	binary	ENCG3 table length
12	C	ENCG3TET (6)	12	binary	ENCG3 table entry triplets
12	C	ENCG3TEO	4	binary	ENCG3 table entry offset
16	10	ENCG3TEL	4	binary	ENCG3 table entry length
20	14	ENCG3TEN	4	binary	ENCG3 table entry number
84	54	ENCG3DEO	4	binary	ENCG3 descriptor entry offset
88	58	ENCG3DEL	4	binary	ENCG3 descriptor entry length
92	5C	ENCG3DEN	4	binary	ENCG3 descriptor entry number
ENCG3 Header Section					
0	0	ENCG3LEN	4	binary	ENCG3 table entry length
4	4	ENCTOKEN	8	EBCDIC	ENCG3 enclave token
12	C	ENCCLX	2	binary	ENCG3 service class index
12	C	ENCPGN	2	binary	ENCG3 performance group
14	E	ENCSRPG	2	binary	ENCG3 subsystem RCLX/RPGN
16	10	ENCNRPG	2	binary	ENCG3 trx name RPGN
18	12	ENCURPG	2	binary	ENCG3 userid RPGN
20	14	ENCCRPG	2	binary	ENCG3 trx class RPGN
22	16	ENCARPG	2	binary	ENCG3 account no RPGN
24	18	ENCPER	1	binary	ENCG3 SC|PG period
25	19	ENCDMN	1	binary	ENCG3 domain
26	1A	ENCG3KFI	1	binary	ENCG3 key field status flags **Bit** **Meaning When Set** **0** ENCG3 key SC/PG has changed **1** ENCG3 key period has changed **2** ENCG3 domain has changed **3-7** Reserved
27	1B	*	9	*	Reserved
36	24	ENCG3EDO	4	binary	ENCG3 offset to EDEG3 element
40	28	ENCG3SMP	4	binary	ENCG3 sample count
44	2C	ENCUSTOT	4	binary	ENCG3 using count Total
48	30	ENCDETOT	4	binary	ENCG3 delay count Total
52	34	ENCIDLES	4	binary	ENCG3 IDLE sample counts
56	38	ENCUNKNS	4	binary	ENCG3 UNKNOWN sample counts
60	3C	ENCUSCPU	4	binary	ENCG3 using count CPU
64	40	ENCDECPU	4	binary	ENCG3 delay count CPU

Offsets		Name	Length	Format	Description
Dec	Hex				
68	44	ENCDECCA	4	binary	ENCG3 delay count CPU capping
72	48	ENCDESTG	4	binary	ENCG3 delay count STOR paging
76	4C	ENCDECOM	4	binary	ENCG3 delay count COM paging
80	50	ENCDEXMM	4	binary	ENCG3 delay count X/M
84	54	ENCDESHP	4	binary	ENCG3 delay count Shared pag
88	58	ENCFLAGS	2	binary	ENCG3 descriptive flags **Bit** **Meaning When Set** **0** ENCG3 dependent enclave **1** ENCG3 original independent enclave **2** ENCG3 foreign independent enclave **3** ENCG3 foreign dependent enclave **4-15** Reserved
90	5A	ENCOASID	2	binary	ENCG3 Owner ASID
92	5C	ENCTOTS	4	binary	ENCG3 multistate samples
96	60	ENCUMCPU	4	binary	ENCG3 using count CPU (multistate samples)
100	64	ENCUMIO	4	binary	ENCG3 using count I/O
104	68	ENCDMCPU	4	binary	ENCG3 delay count CPU (multistate samples)
108	6C	ENCDMIO	4	binary	ENCG3 delay count I/O
112	70	ENCDMQUE	4	binary	ENCG3 delay count queue
116	74	ENCDMCCA	4	binary	ENCG3 delay count capping
120	78	ENCDMSTO	4	binary	ENCG3 delay count storage
124	7C	ENCMIDLE	4	binary	ENCG3 idle count
128	80	ENCMUNKN	4	binary	ENCG3 unknown count
132	84	ENCTCPUT	4	floating	CPU time since creation of enclave
136	88	ENCCPUT	4	floating	CPU time
140	8C	*	8	*	Reserved
148	94	ENCOWSYS	8	EBCDIC	Enclave owner system or blank if not a foreign enclave
156	9C	ENCOWJOB	8	EBCDIC	Enclave owner jobname or blank if not a foreign enclave
164	9C	ENCXTOC	32	EBCDIC	Enclave export token or zero if not a multi-system enclave
196	BC	*	4	*	Reserved
RMF Enclave Descriptor Entry (EDEG3)					
0	0	EDETRXN	8	EBCDIC	EDEG3 transaction program name
8	8	EDEUSER	8	EBCDIC	EDEG3 userid
16	10	EDETRXC	8	EBCDIC	EDEG3 transaction class
24	18	EDENET	8	EBCDIC	EDEG3 netword id
32	20	EDELU	8	EBCDIC	EDEG3 logical unit name

ERBENCG3 - Enclave table

Offsets		Name	Length	Format	Description
Dec	Hex				
40	28	EDEPLAN	8	EBCDIC	EDEG3 plan
48	30	EDEPCKG	8	EBCDIC	EDEG3 package
56	38	EDECNCTN	8	EBCDIC	EDEG3 connection
64	40	EDECOLL	18	EBCDIC	EDEG3 collection
82	52	EDECORR	12	EBCDIC	EDEG3 correlation
94	5E	ECDSUBT	4	EBCDIC	EDEG3 subsystem type
98	62	ECDFCN	8	EBCDIC	EDEG3 function name
106	6A	ECDSUBN	8	EBCDIC	EDEG3 subsystem name
114	72	EDESSPM	255	EBCDIC	EDEG3 subsystem parameter
369	171	EDEACCT	143	EBCDIC	EDEG3 accounting info
512	200	EDE_ PROCEDURENAME	18	EBCDIC	EDEG3 procedure name
530	212	EDE_PERFORM	8	EBCDIC	Perform=value
538	21A	*	2	*	Reserved
540	21C	EDE_PRIORITY	4	binary	Subsystem priority in binary format. Contains X’80000000’ if the subsystem did not provide a priority.
544	220	EDE_ PROCESSNAME	32	EBCDIC	Process name
576	240	*	6	*	Reserved

ERBENTG3 - Enqueue name table

Offsets		Name	Length	Format	Description
Dec	Hex				
ERBENTG3 Header Section					
0	0	ENTENTG3	5	EBCDIC	Acronym ’ENTG3’
5	5	ENTVERG3	1	binary	Control block version X’02’
6	6	ENTHDRLE	1	binary	Length of ENTG3 header
7	7	ENTENTLE	1	binary	Length of one entry
8	8	ENTENTMX	4	binary	Number of table entries
12	C	ENTENTNR	4	binary	Index of last filled entry (Highest possible index is ENTENTMX)
16	10	ENTENTRY (*)	48	EBCDIC	Entries in the ENTG3 table
ERBENTG3 Entry Section					
0	0	ENTENIDX	2	binary	ENQ NAME table entry index
2	2	ENTMAJNA	8	EBCDIC	Major name of this resource
10	A	ENTMINNA	36	EBCDIC	Minor name of this resource
46	2E	ENTSCOPE	1	binary	Scope of this resource **Bit** **Meaning When Set** **0** SYSTEM (When not set: NOSYSTEM) **1** SYSTEMS (When not set: NOSYSTEMS) **2** Reserved **3** GLOBAL (When not set: LOCAL) **4-7** Reserved
47	2F	ENTFLAGS	1	binary	Additional flags **Bit** **Meaning When Set** **0** This resource has suspended jobs. This flag is valid only during data gathering. It is not meaningful within reporter. **1-7** Reserved

ERBGEIG3 - General information table

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	GEIGEIG3	5	EBCDIC	Acronym ‘GEIG3’
5	5	GEIVERG3	1	binary	Control block version X'0B'
6	6	GEILEN	2	binary	Length of this control block (GEIG3)
8	8	*	16	*	Reserved
24	18	GEIVERSN	1	binary	CPU version number
25	19	*	1	*	Reserved
26	1A	GEIFLAG	1	binary	Processor flags **Bit** **Meaning When Set** **0** Service processor architecture supported **1** PR/SM machine **2** Reserved **3** BEG **4** END **5** No collector data **6** Data in GEIGG3 is unpredictable because ERB3GGSS terminated **7** Reserved
27	1B	*	1	*	Reserved
28	1C	GEIMODEL	2	packed	CPU model number (The value is not signed.)
30	1E	GEIIPSID	2	EBCDIC	Installation performance specification (IPS) member suffix
32	20	GEIOPTN	2	EBCDIC	Option (OPT) member suffix
34	22	GEIICSN	2	EBCDIC	Installation control specification (ICS) member suffix
36	24	GEISID	4	EBCDIC	SYSTEM name (SMF system id)
40	28	*	4	*	Reserved
44	2C	GEIAHUIC	4	binary	Highest system unreferenced interval count (HUIC)[1]
48	30	GEIRPOOL_VE	4	floating point	Number of online real storage frames[1]
52	34	GEIRCOMA_VE	4	floating point	Number of real storage COMMON frames[1]
56	38	GEIRSQAA_VE	4	floating point	Number of real storage SQA frames[1]
60	3C	GEIRAFC_VE	4	floating point	Number of available real storage frames[1]
64	40	GEINUCA_VE	4	floating point	Number of nucleus (NUC) frames (real nucleus plus extended storage nucleus frames)[1]
68	44	*	8	*	Reserved
76	4C	GEIEESPL_VE	4	floating point	Number of online extended storage frames[1]
80	50	GEIGAGE_VE	4	floating point	Extended storage migration age[1]
84	54	GEIECOME_VE	4	floating point	Number of extended storage COMMON frames[1]
88	58	GEIEAEC_VE	4	floating point	Number of available extended storage frames[1]

Offsets		Name	Length	Format	Description
Dec	Hex				
92	5C	*	4	*	Reserved
96	60	GEIESQAF_VE	4	floating point	Number of expanded storage SQA frames[1]
100	64	GEIRLPAF_VE	4	floating point	Number of central storage LPA frames[1]
104	68	GEIELPAF_VE	4	floating point	Number of expanded storage LPA frames[1]
108	6C	GEIRCSAF_VE	4	floating point	Number of central storage CSA frames[1]
112	70	GEIECSAF_VE	4	floating point	Number of expanded storage CSA frames[1]
116	74	GEIASMPC	4	binary	Monitor I sample count accumulated per MINTIME used by Monitor III reporter
120	78	GEIASQAO_VE	4	floating point	Number of SQA overflow frames - BEGIN of MINTIME used by Monitor III reporter
124	7C	*	4	*	Reserved
128	80	GEICPM	3	EBCDIC	CP model number
131	83	*	1	*	Reserved
132	84	GEICPUON	2	binary	Snapshot number of online processors at end of the MINTIME
134	86	*	2	*	Reserved
136	88	GEICSASZ	4	binary	IPL Size of CSA below
140	8C	GEISQASZ	4	binary	IPL Size of SQA below
144	90	GEIECSAZ	4	binary	IPL Size of CSA above
148	94	GEIESQAZ	4	binary	IPL Size of SQA above
152	98	GEISTCSA	4	binary	Start of CSA/ECSA tracking (first fullword of TOD)
156	9C	GEISTSQA	4	binary	Start of SQA/ESQA tracking (first fullword of TOD)
160	A0	GEIENCSA	4	binary	End of CSA/ECSA tracking (first fullword of TOD)
164	A4	GEIENSQA	4	binary	End of SQA/ESQA tracking (first fullword of TOD)
168	A8	GEINSCSA	4	binary	Number of CSA samples
172	AC	GEINSSQA	4	binary	Number of SQA samples
176	B0	GEICSAMX	4	binary	Max. allocated CSA below
180	B4	GEISQAMX	4	binary	Max. allocated SQA below
184	B8	GEIECSAX	4	binary	Max. allocated CSA above
188	BC	GEIESQAX	4	binary	Max. allocated SQA above
192	C0	GEICSASP	4	binary	Current allocated CSA below
196	C4	GEISQASP	4	binary	Current allocated SQA below
200	C8	GEIECSAP	4	binary	Current allocated CSA above
204	CC	GEIESQAP	4	binary	Current allocated SQA above
208	D0	GEICSAAV	4	floating point	Accumulated allocated CSA below
212	D4	GEISQAAV	4	floating point	Accumulated allocated SQA below
216	D8	GEIECSAV	4	floating point	Accumulated allocated CSA above
220	DC	GEIESQAV	4	floating point	Accumulated allocated SQA above

ERBGEIG3 - General table

Offsets		Name	Length	Format	Description
Dec	Hex				
224	E0	GEICSACN	4	floating point	Accumulated CSA conv. below
228	E4	GEIECSAN	4	floating point	Accumulated CSA conv. above
232	E8	GEICSACE	4	binary	snapshot CSA conv. below
236	EC	GEIECSAE	4	binary	snapshot CSA conv. above
240	F0	GEICSAAS	4	floating point	Accumulated allocated CSA below
244	F4	GEISQAAS	4	floating point	Accumulated allocated SQA below
248	F8	GEIECSAS	4	floating point	Accumulated allocated CSA above
252	FC	GEIESQAS	4	floating point	Accumulated allocated SQA above
256	100	GEIBATCS	4	floating point	Accumulated allocated CSA below (held by BATCH)
260	104	GEIBATEC	4	floating point	Accumulated allocated SQA below (held by BATCH)
264	108	GEIBATSQ	4	floating point	Accumulated allocated CSA above (held by BATCH)
268	10C	GEIBATES	4	floating point	Accumulated allocated SQA above (held by BATCH)
272	110	GEIASCCS	4	floating point	Accumulated allocated CSA below (held by ASCH)
276	114	GEIASCEC	4	floating point	Accumulated allocated SQA below (held by ASCH)
280	118	GEIASCSQ	4	floating point	Accumulated allocated CSA above (held by ASCH)
284	11C	GEIASCES	4	floating point	Accumulated allocated SQA above (held by ASCH)
288	120	GEIOMVCS	4	floating point	Accumulated allocated CSA below (held by OMVS init.)
292	124	GEIOMVEC	4	floating point	Accumulated allocated SQA below (held by OMVS init.)
296	128	GEIOMVSQ	4	floating point	Accumulated allocated CSA above (held by OMVS init.)
300	12C	GEIOMVES	4	floating point	Accumulated allocated SQA above (held by OMVS init.)
304	130	GEIMTFLG	1	binary	Indicators for the current mintime **Bit** **Meaning When Set** **0** IPS changed during this mintime **1** CSA amounts incomplete in system CAUB **2** SQA amounts incomplete in system CAUB **3** Unexpected VSM error **4** System is in goal mode **5** WLM data not available for this MINTIME **6-7** Reserved
305	131	*	3	*	Reserved
308	134	GEISLID	4	EBCDIC	ID of slip trap

Offsets		Name	Length	Format	Description
Dec	Hex				
312	138	GEIIPLTI	8	EBCDIC	IPL time in TOD format (local time)
320	140	GEIWLMTK	8	EBCDIC	WLM token
328	148	GEISPLXI	8	EBCDIC	Sysplex name
336	150	GEISYSNM	8	EBCDIC	MVS system name
344	158	GEIMAXAS	4	binary	Maximum number of address spaces
348	15C	GEIESPMB	4	floating point	Storage frame movement count: page movement to expanded storage at begin of mintime
352	160	GEIESPME	4	floating point	Storage frame movement count: page movement to expanded storage at end of mintime
356	164	GEIESMRB	4	floating point	Storage frame movement count: migration from expanded storage to auxiliary storage at begin of mintime
360	168	GEIESMRE	4	floating point	Storage frame movement count: migration from expanded storage to auxiliary storage at end of mintime
364	16C	GEIMDL	16	EBCDIC	CPC model identifier
380	17C	GEISEQ	16	EBCDIC	CPC sequence number

[1] Sum of values obtained at each sample. To obtain average values, divide by the number of valid samples (SSHSMPNR).

ERBGGDG3 - Global gatherer data table

Offsets		Name	Length	Format	Description
Dec	Hex				
Control Flow Section					
0	0	GGDGGDG3	5	EBCDIC	Acronym ’GGDG3’
5	5	GGDRMFV	1	EBCDIC	Control block version X’08’
6	6	*	2	*	Reserved
8	8	GGDCRETR	4	binary	Pointer to RETG3 foot print area used for recovery
12	C	GGDMODPT	4	binary	Pointer to GGDMODAR area, array of all gatherer modules
16	10	GGDGOPPT	4	binary	Pointer to GGDGOPT area, gatherer options
20	14	GGDCDCBP	4	binary	Message DCB pointer
24	18	GGDTOFAG	4	binary	Total number of failures of all gatherer modules
28	1C	GGDALLPT	4	binary	Pointer to ERBMFALL module
32	20	GGDCYECB	4	binary	Cycle time ECB **Bit** **Meaning When Set** **0** Cycle time ECB is waited on **1** Cycle time ECB is posted **2-31** Reserved
36	24	GGDSMPNR	4	binary	Sample sequence number
40	28	GGDCBADS	4	binary	Number of consecutive failing samples
44	2C	GGDCBADT	4	binary	Number of consecutive failing for debugging purpose samples threshold value
48	30	GGDCFLAG	4	binary	Gatherer control flags **Bit** **Meaning When Set** **0** Gatherer initializes **1** Gatherer terminates **2** Mintime ended **3** SMF interval ended **4** Not fully initialized because the first set-of-samples will be thrown away **5** New JES2 interface available **6-31** Reserved
52	34	GGDSTDIF	8	EBCDIC	Local Greenwich time
60	3C	GGDCTCYC	8	EBCDIC	Cycle value in TOD format
68	44	GGDCTSTP	8	EBCDIC	Stop time in TOD format
76	4C	GGDCTMNT	8	EBCDIC	Mintime in TOD format
84	54	GGDCTCUC	8	EBCDIC	Begin current cycle in TOD format
92	5C	GGDCTNXC	8	EBCDIC	Begin next cycle TOD format
100	64	GGDCTCUS	8	EBCDIC	Begin current set-of-samples
108	6C	GGDCTNXS	8	EBCDIC	Begin next set-of-samples
Wrap-around Storage Management Section					
116	74	GGDWSHPT	4	binary	Pointer to wrap-around storage header
120	78	GGDWSHTL	4	binary	Total length of wrap-around buffer
124	7C	GGDWSHSP	4	binary	Subpool number of wrap-around buffer

Offsets		Name	Length	Format	Description
Dec	Hex				
Set-of-samples Section					
128	80	GGDSBEGG	8	EBCDIC	Begin time gatherer
136	88	GGDSTBEC	8	EBCDIC	Begin time current sample
144	90	GGDSTENC	8	EBCDIC	End time current sample
152	98	GGDFSSHP	4	binary	Pointer to first SSH control block
156	9C	GGDLSSHP	4	binary	Pointer to last SSH control block
160	A0	GGDCSSHP	4	binary	Pointer to current SSH control block
164	A4	GGDSSHSP	4	binary	Subpool of current set-of-sample area
ENQ Collection Data Space VIAADDR					
168	A8	GGDDSALE	4	EBCDIC	Alet of data space
172	AC	GGDDSORG	4	binary	Origin of data space
Cross-Memory Section					
176	B0	GGDXCELL	4	binary	Pointer to first cell element
180	B4	GGDXCNTR	4	binary	Counter for CDS
184	B8	GGDXETDP	4	binary	Pointer to entry table description
188	BC	GGDJXCPT	4	binary	Pointer to JXCG3 table
192	C0	GGDLXNUM	4	binary	Number of LXs requested
196	C4	GGDLXVAL	4	binary	LX value
200	C8	GGDTKNUM	4	binary	Number of ETs created
204	CC	GGDTKVAL	4	binary	Token returned by ETCRE
HSM Section					
208	D0	GGDMWELE	2	binary	Length of copied MWE part
210	D2	GGDSTALE	2	binary	Length of copied STA part
Data Set Support Section					
212	D4	GGDDSTCB	4	binary	Address of DS subtask TCB
216	D8	GGDDSNPT	4	binary	Pointer to data set names table DSNG3
220	DC	GGDDSSPT	4	binary	Pointer to data set support table DSSG3
224	E0	GGDDSSCT	4	binary	Counter of samples that should have been, but have not been, recorded on DS counted by ERB3GISS
228	E4	GGDDSECB	4	binary	DS stop ECB - DS subtask signals stop complete **Bit** **Meaning When Set** **0** DS stop ECB is waited on **1** DS stop ECB is posted **2-31** Reserved
232	E8	GGDSAVPT	4	binary	Pointer to store subchannel save area
236	EC	GGDIOSPT	4	binary	Pointer to IOSB control block
240	F0	GGDSHBPT	4	binary	Pointer to SCHIB control block
Miscellaneous Section					
244	F4	GGDPMTPT	4	binary	Pointer to performance measurement block
248	F8	GGDCPUVN	1	EBCDIC	CPU version number
249	F9	*	3	*	Reserved

ERBGGDG3 - Global gatherer table

Offsets		Name	Length	Format	Description
Dec	Hex				
252	FC	GGDWSIPT	4	binary	Pointer to wrap-around storage index header
256	100	GGDSID	4	EBCDIC	SMF system-id field
260	104	GGDJESJN	8	EBCDIC	JES jobname
268	10C	GGDJESAS	2	binary	JES ASID number
270	10E	*	2	*	Reserved
272	110	GGDSYNPT	4	binary	Pointer to SYNG3 table
276	114	GGDMNTPT	4	binary	Pointer to temporary OPER MOUNT area
280	118	GGDFLPCT	4	binary	FLPA/EFLPA frames, calculated at initialization
284	11C	*	4	*	Reserved
288	120	GGDASCPT	4	binary	Pointer to ASCG3 table
292	124	GGDCAPPT	4	binary	Binary of common WLM services data capsule
296	128	GGDSPIPT	4	binary	Binary of service policy chain
300	12C	GGDCSVPP	4	binary	Pointer to current SVPG3
304	130	GGDCSRQP	4	binary	Pointer to RQAA capsule
308	134	GGDBDDPT	4	binary	Pointer to diagnose x'204' data area
312	138	GGDCPUXP	4	binary	Pointer to gatherer internal CPUX3 snapshot area
316	13C	GGDCEDAA	4	binary	Pointer to enclave data area
320	140	GGDCEDCC	4	binary	Enclave data cycle count
324	144	GGDCEDFL	4	binary	Enclave data flags **Bit** **Meaning When Set** **0** Enclave data in cycle **1** Enclave data in mintime **2-31** Reserved
328	148	GGDDDNTP	4	binary	Pointer to temporary data set name table
332	14C	GGDBDDPG	4	binary	Length of Diagnose X'204' data in 4K-byte pages
336	150	GGDCSFLG	1	binary	Current set-of-samples control flags **Bit** **Meaning When Set** **0** Write in progress **1** Message ERB321I issued **2-31** Reserved
337	151	GGDCSUCT	3	binary	Current number of users reading the uncompressed data
340	154	GGDCSARL	4	binary	Current set-of-samples area length
344	158	GGDCSARP	4	binary	Current set-of-samples area address (in SUBSSHCF)
348	15C	GGDVRICA	4	binary	VSAMRLS data control area pointer
352	160	GGDOPDCA	4	binary	OMVS process data control area pointer
356	164	*	12	*	Reserved

GGDMODSE - Module Dependent Slot Entry Area

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	GGDMODEN (44)	48	EBCDIC	Module dependent slot

GGDMODAR - Module Dependent Area

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	GGDAUFL1	1	binary	Automatic storage control flag #1 **Bit** **Meaning When Set** **0** Storage assigned **1** Storage must not be freed **2-31** Reserved for user exit routine
1	1	GGDAUFL2	1	binary	Automatic storage control flag #2 **Bit** **Meaning When Set** **0** Area for STA getmained **1-31** Reserved
2	2	GGDAUSBP	2	binary	Subpool number
4	4	GGDAULEN	4	binary	Length of automatic area
8	8	GGDAUPTR	4	binary	Address of automatic area
12	C	GGDMODNA	8	EBCDIC	Module name
20	14	GGDMODAD	4	binary	Entry address of module
24	18	GGDBADMC	1	binary	Consecutive failures this module
25	19	GGDREDNR	1	binary	RED number index
26	1A	GGDREDID	1	binary	RED id
27	1B	GGDMODFL	1	binary	Flag bits for this module **Bit** **Meaning When Set** **0** This module selected to gatherer data **1** This module had permanent error **2** SDUMP requested, continue at retry binary **3-31** Reserved
28	1C	GGDTOFAM	4	binary	Total number of failures of this module
32	20	*	16	*	Reserved

RETG3 - Retry and Footprint Area

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	RETRETG3	5	EBCDIC	Acronym ’RETG3’
5	5	RETRMFV	1	EBCDIC	RETG3 control block version number ’03’x
6	6	RETSTACT	2	binary	Current stack count
8	8	RETSAVE	72	binary	Save area for ERB3GESA
80	50	*	12	*	Reserved

ERBGGDG3 - Global gatherer table

Offsets		Name	Length	Format	Description
Dec	Hex				
92	5C	RETFOOTP	1	binary	Footprint area **Bit** **Meaning When Set** **0** ERB3GINI entered **1** ERB3GTER entered **2** Reserved **3** ERB3GDAS entered **4** ERB3GSTO entered **5** ERB3GJS2 entered **6** ERB3GJS3 entered **7** ERB3GHSM entered
93	5D	*	1	binary	**Bit** **Meaning When Set** **0** ERB3GENQ entered **1** ERB3GMSU entered **2** ERB3GISS entered **3** ERB3GADR entered **4** ERB3GGET entered **5** ERB3GUSR entered **6** ERB3GDSI entered **7** ERB3GGSS entered
94	5E	*	1	binary	**Bit** **Meaning When Set** **0** ERB3GMES entered **1** ERB3GSMF entered **2** ERB3GSIS entered **3** ERB3GSMS entered **4** ERB3GXCF entered **5** ERB3GXCC entered **6** ERB3GMGP entered **7** ERB3GCSR entered
95	5F	*	1	binary	**Bit** **Meaning When Set** **0** ERB3GIXC entered **1** ERB3GIXI entered **2** ERB3GSTH entered **3** ERB3GCFS entered **4** ERB3GCFC entered **5** ERB3GCFI entered **6** ERB3GMRC entered **7** ERB3GMRG entered
96	60	*	1	binary	**Bit** **Meaning When Set** **0** ERB3GSCM entered **1** ERB3GRQA entered **2** ERB3GSMG entered **3** ERB3GSIG entered **4** ERB3GEN0 entered **5** ERB3GEN1 entered **6** ERB3GEN2 entered **7** ERB3GEN3 entered

Offsets		Name	Length	Format	Description
Dec	Hex				
97	61	*	1	binary	**Bit** **Meaning When Set** **0** ERB3GEN5 entered **1** ERB3GJSX entered **2** ERB3GSM2 entered **3** ERB3GHFS entered **4** ERB3GCTC entered **5** ERB3GVRI entered **6** ERB3GOPD entered **7** Reserved
98	62	*	2	*	Reserved
100	64	RETFLAG2	1	binary	Retry flag #2 **Bit** **Meaning When Set** **0** Recursion **1** XMEM established **2** ENQ environment established **3** Message dataset opened **4** ERB3GXIT was entered **5** ERB425I issued for JES2 **6** ERB425I issued for JES3 **7** ERB425I issued for HSM
101	65	RETFLAG3	1	binary	Retry flag #3 **Bit** **Meaning When Set** **0** ALESERV issued for data space, set on by ERB3GENQ, set off by ERB3GTER **1-3** Reserved **4** Store subchannel entered **5-7** Reserved
102	66	RETFLAG4	1	binary	Retry flag #4 **Bit** **Meaning When Set** **0** User exit routine loaded **1** ERB3GDSI entered via error recovery module erb3gesa **2** ERB3GTEQ entered **3** ERB3GXTE entered **4** Cancel TTIMER request **5** ERBSMFI loaded **6-7** Reserved
103	67	*	29	*	Reserved
132	84	RETSTAAR (10)	96	binary	Retry stack area

RETSTACK - Retry Stack Element

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	RETTIMBE	8	EBCDIC	Time stamp begin
8	8	RETADDR	4	binary	Pointer to retry routine or zero
12	C	RETCOUNT	4	binary	Retry count for this CSECT
16	10	RETAMEMP	4	binary	Pointer to module dependent slot

ERBGGDG3 - Global gatherer table

Offsets		Name	Length	Format	Description
Dec	Hex				
20	14	RETFLAG1	2	binary	Retry flag #1 **Bit** **Meaning When Set** **0-1** Reserved **2** Issue message ERB280I **3** Issue message ERB268I **4** Issue message ERB269I **5-15** Reserved
22	16	RETRUBFL	2	binary	Select registers flag
24	18	RETREGSA	64	binary	Save area for RUB
88	58	RETLSIDX	4	binary	Offset of link stack entry to recover
92	5C	*	4	*	Reserved

GGDGOPT - Options Area

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	GGDGOCYC	4	binary	Gatherer option CYCLE
4	4	GGDGOSTP	4	binary	Gatherer option STOP **Bit** **Meaning When Set** **0** NOSTOP option active **1-31** Reserved
8	8	GGDGOSYN	4	binary	Gatherer option SYNCH **Bit** **Meaning When Set** **0** NOSYNCH option active **1-31** Reserved
12	C	GGDGOMNT	4	binary	Gatherer option MINTIME
16	10	*	3	*	Reserved
19	13	GGDGOCLA	1	EBCDIC	Gatherer option SYSOUT: sysout class alphanumeric value
20	14	GGDGORES	2	binary	Resource selected options **Bit** **Meaning When Set** **0** Resource Processor selected **1** Resource DASD selected **2** Resource Storage selected **3** Resource JES2 selected **4** Resource JES3 selected **5** Resource HSM selected **6** Resource ENQ selected **7** Resource Oper MSG selected **8** Resource Oper MOUNT selected **9** Resource XCF selected **10** Resource OMVS selected **11-14** Reserved **15** Resource User selected

Offsets		Name	Length	Format	Description
Dec	Hex				
22	16	GGDGOFL1	1	binary	Flag byte #1 **Bit** **Meaning When Set** **0** Performance measurement active **1** Data set support selected **2** IPM measurement requested **3** CFDETAIL requested **4** CACHE requested **5** VSAMRLS requested **6** OPD requested **7** Reserved
23	17	*	1	*	Reserved
24	18	GGDJESN	4	EBCDIC	Name of JES subsystem chosen
28	1C	GGDGOWHL	4	binary	Value dataset option WHOLD
32	20	GGDGOWST	4	binary	Value of option WSTOR
36	24	GGDGOSOF	4	binary	Offset of synch point from the full hour used by gatherer. Units are full seconds.
40	28	GGDGOCCU	4	binary	Pointer to bit array for selected cache SSIDs
44	2C	*	32	*	Reserved

ERBPGPER - Performance group period table

Offsets		Name	Length	Format	Description
Dec	**Hex**				
PGPER Header Section:					
0	0	PGPNAME	5	EBCDIC	Acronym 'PGPER'
5	5	PGPVERS	1	binary	Control block version X'02'
6	6	*	1	*	Reserved
8	8	PGPHDRLN	2	binary	Length of header
10	A	PGPENTLN	2	binary	Length of one entry
12	*	*	4	binary	Reserved
16	10	PGPSUBPN	2	binary	Subpool number
18	12	PGPTOTLN	2	binary	Total length of PGPER table
20	14	PGPHIPG	2	binary	Highest PG number
22	16	PGP#PGP	2	binary	Number of PGP entries
24	18	PGPSRMCT	2	binary	SRM command count. This count is incremented by 1 every time the gatherer (Monitor I or common collector or Monitor III) sees a SET IPS or SET ICS command. Source is STGSSRMC.
26	1A	*	38	*	Reserved
64	40	PGPENTRY	20	EBCDIC	PGPER entry
PGPER Table Entry Section:					
0	0	PGPPGN	2	binary	PG number
2	2	PGPERD	2	binary	PG period number
4	4	PGPDMN	2	binary	Domain number
6	6	PGPFLAG1	2	binary	PGP flags **Value** **Meaning** **0** Performance Group Report **1-15** Reserved
8	8	PGPET	4	binary	Total elapsed time for all transactions that ended in the performance period group. Does not include queued time. In units of 1024 microseconds.
12	C	PGPQT	4	binary	Total time spent on JES or APPC queues by all transactions that ended in the performance period group. In units of 1024 microseconds.
16	10	PGPTRN	4	binary	The number of the transactions that ended in the performance group period.

ERBRCDG3 - Resource collection data

Resource Collection Data Header

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	RCDACRO	5	EBCDIC	Acronym ’RCDG3’
5	5	RCDVERS	1	binary	Control block version X'04'
6	6	RCDHLEN	2	binary	Size of RCDHDR
8	8	RCDSIZ	4	binary	Size of all resource collection data. This includes RCDHDR, RCDBMAP, RCDG3, RCDPD, RCDRD and RCDSD.
12	C	RCDPNAM	8	EBCDIC	Policy name
20	14	RCDPTM	8	binary	Local time policy was activated (TOD format)
28	1C	RCDNTVL	4	binary	Current sample interval (in milliseconds). This is the frequency with which WLM samples delays reported in the RCAA.
32	20	RCDNTV#	4	binary	Total number of times WLM sampling code ran. A monitor issuing successive calls to IWMRCOLL should not assume that WLM sampling code ran at the interval specified by RCDNTVL between its calls. This field can be used to translate sampled state data into actual percentages of time.
36	24	RCDMSC#	2	binary	Maximum possible number of service classes according to SVPOL service class array
38	26	RCDMRC#	2	binary	Maximum possible number of report classes according to SVPOL report class array
40	28	RCDMPD#	2	binary	Maximum possible number of service or report class period entries according to SVPOL.
42	2A	RCDMRD#	2	binary	Maximum possible number of response time distribution buckets according to number of periods with response time goals
44	2C	RCDBMPL	2	binary	Length of an entry in the response time distribution mapping array
46	2E	RCDBMP#	2	binary	Number of response time distribution buckets
48	30	RCDBMPO	4	binary	Offset from begin of RCDHDR to response time distribution mapping array (RCDBMAP)
52	34	RCDSCAL	2	binary	Length of one RCDG3 workload activity entry in the RCDSCOF array
54	36	RCDSCA#	2	binary	Number of entries in RCDSCOF array. This is the number of service classes returned in IWMSVPOL by IWMPQRY.
56	38	RCDSCOF	4	binary	Offset from begin of RCDHDR to array of RCDG3 entries. These entries represent service classes.
60	3C	RCDRCAL	2	binary	Length of one RCDG3 workload activity entry in the RCDRCOF array
62	3E	RCDRCA#	2	binary	Number of entries in RCDRCOF array. This field is the number of report classes returned in IWMSVPOL by IWMPQRY.

ERBRCDG3 - Resource collection data

Offsets		Name	Length	Format	Description
Dec	Hex				
64	40	RCDRCOF	4	binary	Offset from begin of RCDHDR to array of RCDG3 entries
68	44	RCDPDAL	2	binary	Length of one RCDG3 period entry in the RCDPD array
70	46	RCDPDA#	2	binary	Number of entries in the RCDPD array
72	48	RCDPDAO	4	binary	Offset from begin of RCDHDR to begin of RCDPD array
76	4C	RCDRDAL	2	binary	Length of one RCDG3 response time bucket entry in the RCDRD array
78	4E	RCDRDA#	2	binary	Number of entries in the RCDRD array
80	50	RCDRDAO	4	binary	Offset from begin RCDHDR to begin of RCDRD array
84	54	RCDSDAL	2	binary	Length of one RCDG3 subsystem delay data entry in the RCDSD array
86	56	RCDSDA#	2	binary	Number of entries in the RCDSD array
88	58	RCDSDAO	4	binary	Offset from begin of RCDHDR to begin of RCDSD array
92	5C	RCDSUBP	2	binary	Subsystem phase count X'0002'
94	5E	*	2	*	Reserved
Response Time Distribution Map Array					
0	0	RCDBENT	4	binary	Response time distribution bucket mappings. Each word defines a maximum % of a goal (ie. 50, 70, 100, etc.) When used in conjunction with an RCDDENT, a monitor product can show the number of transactions that completed in a percentage of a goal. The last entry in the array contains X'FFFFFFFF'. This indicates that this bucket includes all transactions that completed with longer response times than the previous bucket.
Resource Collection Data Entry					
0	0	RCDTYPE	1	binary	What this RCDG3 entry represents **Bit** **Meaning When Set** **0** Service class **1** Report class **2-7** Reserved
1	1	RCDFLGS	1	binary	Class data availability flags **Bit** **Meaning When Set** **0** Service classes served **1-7** Reserved
2	2	RCDCLX	2	binary	Index into the service class or report class list mapped by SVPCD and SVPHD, respectively (service policy information)
4	4	RCDMP#	1	EBCDIC	Maximum possible number of periods for this RCDG3.
5	5	RCDMB#	1	EBCDIC	Maximum possible number of response time distribution buckets for this RCDG3.

Offsets		Name	Length	Format	Description
Dec	Hex				
6	6	RCDPD#	2	binary	Number of period data entries for this RCDG3 entry
8	8	RCDPDI	4	binary	Index into RCDG3 period entry array
12	C	RCDFRX	2	binary	Index to first RT-distribution bucket of this class
14	E	RCDCR#	2	binary	Number of buckets for this class
16	10	RCDFSX	2	binary	Index to first subsystem delay data entry of this class
18	12	RCDCS#	2	binary	Number of subsystem delay data entries for this class
Resource Collection Data - Period Entry					
0	0	RCDPFLGS	1	binary	Data availability flags **Bit** **Meaning When Set** **0** Resource consumption data **1** Response time data **2** General execution delay data **3-7** Reserved
1	1	RCDPFLG1	1	binary	Report class period flags **Bit** **Meaning When Set** **0** Heterogeneous report class period **1-7** Reserved
2	2	RCDPLSC	2	binary	Index of the service class that last contributed to this report class. For homogeneous report class periods, this service class period's goal has to be used to format the response time distribution for ended transactions reported in this report class. Zero for a service class entry.
4	4	RCDPERI	1	binary	Period number
5	5	RCDRD#	1	binary	Number of entries in the response time distribution bucket array (RCDRD) that belong to this period or zero
6	6	RCDRDI	2	binary	Index into response time distribution bucket array. This field will be zero when there are no response time goals specified.
8	8	RCDSD#	2	binary	Number of entries in the subsystem work manager delay array (RCDSD) that belong to this period or zero
10	A	RCDSDI	2	binary	Index into subsystem work manager delay data array. Zero means, there is no subsystem work manager delay data for this period.
12	C	RCDCPU	8	binary	Total CPU service units for this period
20	14	RCDSRB	8	binary	Total SRB service units for this period
28	1C	RCDRCP	4	binary	Count of transaction completions for this period. This field also includes transaction completions reported by subsystem work managers via the IWMRPT service.

ERBRCDG3 - Resource collection data

Offsets		Name	Length	Format	Description
Dec	Hex				
32	20	RCDARCP	4	binary	Count of transactions that completed abnormally as reported by subsystem work managers. This value is not part of RCDRCP and should not be used for response time calculations.
36	24	RCDNCP	4	binary	Count of transactions that completed their execution phase as reported by subsystem work managers via the IWMMNTFY service.
40	28	RCDANCP	4	binary	Count of transactions that completed their execution phase abnormally as reported by subsystem work manager. This value is not part of RCANCP and should not be used for execution response time calculations.
44	2C	RCDTET	8	binary	Total transaction elapsed time (in 1024-microsecond units)
52	34	RCDXET	8	binary	Total transaction execution time (in 1024-microsecond units), available in goal mode only
60	3C	RCDCUSE	4	binary	Total using samples
64	40	RCDTOTD	4	binary	Total delay samples used in SRM's execution velocity calculation
68	44	RCDQDT	8	binary	Queue delay time (in 1024-microsecond units)
76	4C	RCDADT	8	binary	Resource affinity delay time (in 1024-microsecond units)
84	54	RCDCVT	8	binary	JCL conversion delay time (in 1024-microsecond units)
92	5C	RCDIQT	8	binary	Ineligible queue time (in 1024-microsecond units)
Resource Collection Data - Response Time Distribution Array					
0	0	RCDDENT	4	binary	An entry in the RCDG3 response time distribution array. Each entry in the array contains the number of transactions that completed in the time period represented by that entry. When used with the response time distribution bucket mapping (RCDBMAP), monitors can construct a distribution of completions verses goals specified.
Resource Collection Data - Subsystem Work Manager Delays					
0	0	RCDSTYP	4	EBCDIC	Subsystem type, as used in the classification rules specified in the WLM administrative application
4	4	RCDEFLG	1	binary	Flags **Bit** **Meaning When Set** **0** Represents states sampled in the begin-to-end phase of a transaction **1** Represents states sampled in the execution phase of a transaction **2-7** Reserved
5	5	*	3	*	Reserved

Offsets		Name	Length	Format	Description
Dec	Hex				
8	8	RCDESS#	4	binary	Total number of transaction states sampled in the work phase specified by RCDEFLG
12	C	RCDACTV	4	binary	Total number of active state samples. Active indicates that there is a program executing on behalf of the work request, from the perspective of the work manager. This does not mean that the program is active from the base control program's perspective.
16	10	RCDRDY	4	binary	Total number of ready state samples. Ready indicates that there is a program ready to execute on behalf of the work request described by the monitoring environment, but the work manager has given priority to another work request.
20	14	RCDIDL	4	binary	Total number of idle state samples. Idle indicates that no work request is avaialable to the work manager that is allowed to run.
24	18	RCDWLOK	4	binary	Total number of waiting for lock state samples
28	1C	RCDWIO	4	binary	Total number of waiting for I/O state samples. Waiting for I/O indicates that the work manager is waiting for an activity related to an I/O request. This may be an actual I/O operation or some other function associated with the I/O request.
32	20	RCDWCON	4	binary	Total number of waiting for conversation state samples. Waiting for conversation may have been used in conjunction with the WLM service IWMMSWCH to identify where the recipient of the conversation is located. In this case, only the switched state will be recorded.
36	24	RCDWDST	4	binary	Total number of waiting for distributed request state samples. Waiting for distributed request indicates a high level that some function or data must be routed prior to resumption of the work request. This is to be contrasted with waiting for conversation, which is a low level view of the precise resource that is needed. A distributed request could involve waiting on a conversation as part of its processing.
40	28	RCDWSL	4	binary	Waiting for a session to be established locally, ie. on the current MVS image
44	2C	RCDWSN	4	binary	Waiting for a session to be established somewhere in the network
48	30	RCDWSS	4	binary	Waiting for a session to be established somewhere in the sysplex
52	34	RCDWTMR	4	binary	Waiting for a timer
56	38	RCDWO	4	binary	Waiting for another product
60	3C	RCDWMSC	4	binary	Waiting for unidentified resource, possibly among another more specific category, but which may not be readily determined

ERBRCDG3 - Resource collection data

Offsets		Name	Length	Format	Description
Dec	Hex				
64	40	RCDSSL	4	binary	State representing transactions for which there are logical continuations on this MVS image. Subsystem work managers might set this state when they function ship a transaction to another component within the same MVS image.
68	44	RCDSSS	4	binary	State representing transactions for which there are logical continuations on another MVS image in the sysplex. Subsystem work managers might set this state when they function ship a transaction to another component on another MVS image within the sysplex.
72	48	RCDSSN	4	binary	State representing transactions for which there are logical continuations somewhere within the network. Subsystem work managers might set this state when they function ship a transaction to another component within the network.
76	4C	*	16	*	Reserved
92	5C	RCDWNL	4	binary	Total number of state samples reflecting waiting for new latch
96	60	RCDACTA	4	binary	Total number of active application state samples. Active application indicates a program is executing on behalf of the work request, from the perspective of the work manager. This does not mean that the program is active from the base control program's perspective.
100	64	RCDWSSL	4	binary	Total number of waiting for an SSL thread samples
104	68	RCDWRET	4	binary	Total number of waiting for a regular thread samples
108	6C	RCDWREW	4	binary	Total number of waiting for a registration to a work table samples
112	70	RCDWTY1	4	binary	Total number of waiting for resource type 1 samples
116	74	RCDWTY2	4	binary	Total number of waiting for resource type 2 samples
120	78	RCDWTY3	4	binary	Total number of waiting for resource type 3 samples
124	7C	RCDWTY4	4	binary	Total number of waiting for resource type 4 samples
128	80	RCDWTY5	4	binary	Total number of waiting for resource type 5 samples
132	84	*	40	*	Reserved

ERBREDG3 - Resource data record

Offsets		Name	Length	Format	Description
Dec	Hex				
ERBREDG3 Header Section					
0	0	REDREDID	1	binary	RED ID **Bit** **Meaning When Set** **0-1** Reserved **2-7** Resource ID
1	1	REDFLAG1	1	binary	RED flags **Bit** **Meaning When Set** **0** This resource is invalid **1** USE records available **2** WAIT records available **3-7** Reserved
2	2	*	2	*	Reserved
4	4	REDFUWDO	4	binary	Offset to first USE/WAIT record
8	8	REDUWDL1	1	binary	Short length of ENQ UWD record (without SYSTEM/JOBNAME)
9	9	REDUWDL2	1	binary	Total length of ENQ UWD record (with SYSTEM/JOBNAME)
10	A	REDUSERN	2	binary	Number of user-exit records
ERBREDG3 Array Entry					
0	0	REDENTRY (10)	12	EBCDIC	Entry in RED array

ERBSHDG3 - Sample header

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	SHDSHDG3	5	EBCDIC	Acronym 'SHDG3'
5	5	SHDRMFV	1	binary	Control block version X'02'
6	6	SHDLEN	1	binary	Length of SHDG3
7	7	SHDFLAG1	1	binary	Sample flag **Bit** **Meaning When Set** **0** Sample is invalid **1-7** Reserved
8	8	SHDPREVO	4	binary	Offset to previous sample. This field contains the offset within the Monitor III data gatherer areas. The Monitor III reporter module changes the offset to a pointer after the data have been moved to the reporter's address space
12	C	SHDNEXTO	4	binary	Offset to next sample. This field contains the offset within the Monitor III data gatherer areas. The Monitor III reporter module changes the offset to a pointer after the data have been moved to the reporter's address space
16	10	SHDREDOF	4	binary	Offset to first RED record
20	14	SHDREDNR	2	binary	Number of RED records
22	16	SHDREDLE	2	binary	Length of one REDG3 entry
24	18	*	6	*	Reserved
30	1E	SHDUWDNR	2	binary	Number of Use/Wait records
32	20	*	16	*	Reserved

ERBSSHG3 - MINTIME set of samples header

Offsets		Name	Length	Format	Description
Dec	Hex				
Set of Samples Header Section:					
0	0	SSHSSHG3	5	EBCDIC	Acronym ‘SSHG3’
5	5	SSHRMFV	1	binary	Control block version X'0C'
6	6	SSHLEN	2	binary	Length of set of samples header (SSHG3)
8	8	SSHRMFVN	3	EBCDIC	RMF version number
11	B	SSHFLAG1	1	binary	Flag byte **Bit** **Meaning** **0** Data are compressed **1** Goal mode data **2-7** Reserved
12	C	*	24	*	Reserved
36	24	SSHSHDFO	4	binary	Offset of first sample header from ERBSSHG3
40	28	SSHSHDLO	4	binary	Offset of last sample header from ERBSSHG3
44	2C	SSHTOTLE	4	binary	Total length for this set of samples (including the set of samples header)
48	30	*	8	*	Reserved
56	38	SSHSMPNR	4	binary	Number of valid samples
60	3C	SSHTIBEG	8	binary	Begin time for this set of samples
68	44	SSHTIEND	8	binary	End time for this set of samples
76	4C	*	16	*	Reserved
92	5C	SSHASIO	4	binary	Offset of the ASID table from ERBSSHG3
96	60	*	12	*	Reserved
108	6C	SSHDVTO	4	binary	Offset of the DVT table from ERBSSHG3
112	70	*	8	*	Reserved
120	78	SSHENTO	4	binary	Offset of the ENT table from ERBSSHG3
124	7C	*	12	*	Reserved
136	88	SSHPMTO	4	binary	Offset to PTMG3
140	8C	*	16	*	Reserved
148	94	SSHGEIO	4	binary	Offset of the general information table (GEIG3) from ERBSSHG3
152	98	SSHIOML	1	binary	Processor type on which data was created **Value** **Meaning** **X'03'** 9672, zSeries 900
153	99	SSHEFLAG	1	binary	Extended storage indicators **Bit** **Meaning When Set** **0** Extended storage installed **1-7** Reserved

ERBSSHG3 - Samples header

Offsets		Name	Length	Format	Description
Dec	Hex				
154	9A	SSHPRFGS	2	binary	Processor flags **Bit** **Meaning When Set** **0** ES/Connection Channel enabled **1** ES/Connection Director configured **2-7** Reserved
156	9C	SSHGOCYC	4	binary	Gatherer CYCLE option
160	A0	SSHGOSTP	4	binary	Gatherer STOP option. (If the first bit is set to 0, NOSTOP is in effect.)
164	A4	SSHGOSYN	4	binary	Gatherer SYNC option. (If the first bit is set to 0, NOSYNC is in effect.)
168	A8	SSHGOMNT	4	binary	Gatherer MINTIME option
172	AC	*	3	*	Reserved
175	AF	SSHGOCLA	1	EBCDIC	Gatherer SYSOUT class option
176	B0	*	4	*	Reserved
180	B4	SSHJESN	4	EBCDIC	Name of JES subsystem
184	B8	SSHGOWHL	4	binary	Gatherer DATASET WHOLD suboption
188	BC	SSHGOWST	4	binary	Gatherer WSTOR option
192	C0	*	40	*	Reserved
232	E8	SSHSTDIF	8	binary	Difference between local time and Greenwich Mean Time where the difference equals local time minus Greenwich Mean Time
240	F0	SSHHSMJN	8	EBCDIC	Jobname of HSM subsystem
248	F8	SSHHSMAS	2	binary	ASID number of HSM subsystem
250	FA	SSHJESJN	8	EBCDIC	Jobname of JES subsystem
258	102	SSHJESAS	2	binary	ASID number of JES subsystem
260	104	SSHPGPO	4	binary	Offset to PGPER control block. This field contains the offset when the data is within the wrap around buffer.
264	108	*	4	*	Reserved
268	10C	SSHCSRO	4	binary	Offset to CSR table. This field contains the offset when the data are within the wrap around buffer.
272	110	SSHJLCYC	4	binary	Time-offset when the last cycle was gathered, expressed in CYCLE time units.
276	114	*	4	*	Reserved
280	118	SSHRCDO	4	binary	Offset to RCDG3
284	11C	SSHCPUO	4	binary	Offset to CPUG3
288	120	SSHIPLTI	8	binary	IPL time in TOD format
296	128	SSHWLMTK	8	binary	WLM token
304	130	SSHENCO	4	binary	Offset to ENCG3

Offsets		Name	Length	Format	Description
Dec	Hex				
308	134	SSHSM2O	4	binary	Offset to SM2G3
312	138	SSHDDNO	4	binary	Offset to DDNG3
316	13C	SSHCFIO	4	binary	Offset to CFIG3
320	140	SSHCATO	4	binary	Offset to CATG3
324	144	SSHVRIO	4	binary	Offset to VRIG3
328	148	SSHOPDO	4	binary	Offset to OPDG3
332	14C	*	4	*	Reserved

ERBSVPG3 - Service policy

Offsets		Name	Length	Format	Description
Dec	Hex				
Service Policy Header					
0	0	SVPNAM	5	EBCDIC	Acronym 'SVPG3'
5	5	SVPDVN	1	binary	Control block version X'01'
6	6	SVPDIL	2	binary	Length of header section
8	8	SVPDLE	4	binary	Total length of the active service policy data structure
12	C	SVPTIB	8	EBCDIC	Begin time in TOD. Time, policy has been activated
20	14	SVPTIE	8	EBCDIC	End time in TOD. Time, policy has been deactivated or compatibility mode has been activated
28	1C	SVPDPO	4	binary	Offset to the service policy definition section
32	20	SVPDPL	2	binary	Length of the policy entry in the policy section
34	22	*	2	*	Reserved
36	24	SVPDWO	4	binary	Offset to the workload definition section
40	28	SVPDWC	2	binary	Number of workload entries in the workload definition section
42	2A	SVPDWL	2	binary	Length of each workload entry
44	2C	SVPDCO	4	binary	Offset to the service class definition section
48	30	SVPDCC	2	binary	Number of service class entries in the service class definition section
50	32	SVPDCL	2	binary	Length of each service class definition entry
52	34	SVPDZO	4	binary	Offset of service class period entries
56	38	SVPDZC	2	binary	Number of service class periods
58	3A	SVPDZL	2	binary	Length of each service class period entry
60	3C	SVPDRO	4	binary	Offset to the report class definition section
64	40	SVPDRC	2	binary	Number of report class entries in the report class definition section
66	42	SVPDRL	2	binary	Length of each report class definition entry
68	44	SVPDGO	4	binary	Offset to the resource group definition section
72	48	SVPDGC	2	binary	Number of resource group entries in the resource group definition
74	4A	SVPDGL	2	binary	Length of each resource group definition entry
76	4C	*	52	*	Reserved
Service Policy					
0	0	SVPNSP	8	EBCDIC	Service policy name
8	8	SVPDSP	32	EBCDIC	Service policy description
40	28	SVPTPA	8	EBCDIC	Time/date (TOD format) of policy activation
48	30	SVPIPU	8	EBCDIC	Userid of the system operator or service administrator who activated the service policy

Offsets		Name	Length	Format	Description
Dec	Hex				
56	38	SVPSNA	8	EBCDIC	Name of the system on which policy activation was initiated
64	40	SVPSEQ	4	binary	Classification sequence number
68	44	SVPASN	4	binary	Activation sequence number
72	48	SVPIDN	8	EBCDIC	Name of the service definition from which the service policy was extracted
80	50	SVPTDI	8	EBCDIC	Time/date (TOD format) that the service definition was installed
88	58	SVPIDU	8	EBCDIC	Userid of the service administrator who installed the service definition
96	60	SVPIDS	8	EBCDIC	Name of the system on which the service definition was installed
104	68	SVPIDD	32	EBCDIC	Description of service definition from which the service policy was extracted
136	88	SVPCPU	4	binary	CPU service coefficient *10000 - the number by which accumulated CPU service units will be multiplied (weighted)
140	8C	SVPIOC	4	binary	I/O service coefficient * 10000 - the number by which accumulated I/O service units will be multiplied (weighted)
144	90	SVPMSO	4	binary	Storage service coefficient (MSO) * 10000 - the number by which accumulated storage service units will be multiplied (weighted)
148	94	SVPSRB	4	binary	SRB service coefficient * 10000 - the number by which accumulated SRB service units will be multiplied (weighted)
152	98	SVPECP	4	EBCDIC	EBCDIC representation of CPU service coefficient
156	9C	SVPEIO	4	EBCDIC	EBCDIC representation of I/O service coefficient
160	A0	SVPEMS	8	EBCDIC	EBCDIC representation of Storage service coefficient
168	A8	SVPESR	4	EBCDIC	EBCDIC representation of SRB service coefficient
172	AC	*	4	*	Reserved
Workload Information					
0	0	SVPWNM	8	EBCDIC	Workload name
8	8	SVPWDE	32	EBCDIC	Workload description
Service Class Information					
0	0	SVPCNM	8	EBCDIC	Service class name
8	8	SVPCDE	32	EBCDIC	Service class description
40	28	SVPCWN	8	EBCDIC	Name of the workload this service class is associated with
48	30	SVPCRN	8	EBCDIC	Name of the resource group this service class is associated with - blanks if no resource group association

ERBSVPG3 - Service policy

Offsets		Name	Length	Format	Description
Dec	Hex				
56	38	SVPCPO	4	binary	Offset of service class period entries for this service class
60	3C	SVPCPN	2	binary	Number of service class periods for this service class
62	3E	*	2	*	Reserved
64	40	SVPCGI	4	binary	Resource group index - the index of the resource group entry in SVPRG of the resource group to which this service class belongs
68	44	SVPCWI	4	binary	Workload index - the index of the workload entry in SVPWD of the workload to which this service class belongs
72	48	SVPCRC	4	binary	Number of periods with response time goals specified
Service Class Period Information					
0	0	SVPTYP	4	binary	Goal type indicators **Bit** **Meaning When Set** **0** Percentile response time goal **1** Average response time goal **2** Velocity goal **3** Discretionary goal **4** System goal **5-7** Reserved
4	4	*	1	*	Reserved
5	5	SVPRTU	1	binary	Response time unit indicator indicating the units in which SVPVAL is expressed
6	6	SVPPER	2	binary	Goal percentile value
8	8	SVPIMP	2	binary	Importance level ranging from 1 to 5 where 1 is most important
10	A	*	2	*	Reserved
12	C	SVPVAL	4	binary	Response time goal or velocity goal. Zero if discretionary or system goal or no goal defined.
16	10	SVPDUR	4	binary	Service class period duration in service units, or zero for last period
Resource Group Information					
0	0	SVPGNM	8	EBCDIC	Resource group name
8	8	SVPGDE	32	EBCDIC	Resource group description
40	28	SVPGMN	4	binary	If SVPMNS = 1, this field contains the minimum capacity in unweighted CPU service units per second, otherwise this field is 0
44	2C	SVPGMX	4	binary	If SVPMXS = 1, this field contains the maximum capacity in unweighted CPU service units per second, otherwise this field is 0
48	30	SVPGLT	4	binary	Indicators **Bit** **Meaning When Set** **0** Maximum capacity was specified **1** Minimum capacity was specified **2-7** Reserved

Offsets		Name	Length	Format	Description
Dec	Hex				
Report Class Information					
0	0	SVPRNM	8	EBCDIC	Report class name
8	8	SVPRDE	32	EBCDIC	Report class description

ERBUWDG3 - USE/WAIT record

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	UWDUWRID	1	binary	USE/WAIT record id **Bit** **Meaning When Set** **0** WAIT record **1** USE record **2-7** Resource identification
1	1	UWDASID	2	binary	Address space (ASIG3) table index
Extended Data for PROC Section (See resource id in UWDUWRID):					
3	3	UWDFLAGP	1	binary	Flag for processor delay types **Bit** **Meaning When Set** **0** Resource was used by enclaves **1-7** Reserved
Extended Data for DEV Section (See resource id in UWDUWRID):					
3	3	UWDDEVNR	2	binary	Device table (DVTG3) index
Extended Data for STOR Section See resource id in UWDUWRID):					
3	3	UWDPDEVR	2	binary	Paging device DVTG3 index
5	5	UWDFLAGS	1	binary	Flag for storage status **Bit** **Meaning When Set** **0** Delayed for LOCAL request **1** Delayed for SWAP IN request **2** Delayed for COMMON request **3** Delayed for VIO request **4** Space type LOCL **5** Reserved **6** Space type COMM **7** Space type PLPA
Extended Data for JES2/JES3 section (See resource id in UWDUWRID):					
3	3	UWDJESFU	2	binary	JES2/JES3 function code For a list of JES function codes, refer to the description of the JES Delays report in *z/OS RMF Report Analysis*.
5	5	UWDJS3MO	1	binary	JES3 modification code
Extended Data for HSM Section (See resource id in UWDUWRID):					
3	3	UWDHSMFU	1	binary	HSM function code For a list of HSM function codes, refer to the description of the HSM Delays report in *z/OS RMF Report Analysis*.
4	4	UWDHSMMO	1	binary	HSM modification code
Extended Data for ENQ Section (See resource id in UWDUWRID):					
3	3	UWDENTID	2	binary	ENQUEUE name table (ENTG3) index

Offsets		Name	Length	Format	Description
Dec	Hex				
5	5	UWDFLAGE	1	binary	ENQUEUE flags **Bit** **Meaning When Set** **0** OFF=Request is EXCLUSIVE ON=Request is SHARED **1** ON=Request from another system. (Fields UWDSYSNA/UWDJOBNA are valid) **2** Server name present **3-7** Reserved
6	6	UWDSASID	2	binary	Server address space analysis index. Valid if Bit 2 of UWDFLAGE is set.
6	6	UWDSYSNA	8	EBCDIC	System name of requestor. Valid if Bit 1 of UWDFLAGE is set.
14	E	UWDJOBNA	8	EBCDIC	Job name of requestor. Valid if Bit 1 of UWDFLAGE is set.
Extended Data for MESSAGE section (See resource id in UWDUWRID):					
3	3	UWDOREID	4	EBCDIC	Reply number
Extended Data for MOUNT section (See resource id in UWDUWRID):					
3	3	UWDDEVIN	2	binary	DVTG3 table index
Extended Data for XCF section (See resource id in UWDUWRID):					
3	3	UWDXCDEV	4	EBCDIC	Device number of path on which the message is pending
7	7	UWDXCMAS	2	binary	ASID of member sending message
9	9	UWDXCHAS	2	binary	Name of ASID that initiated message out request

ERBXMHG3 - Moved samples header control block

Offsets		Name	Length	Format	Description
Dec	Hex				
0	0	XMHXMHG3	5	EBCDIC	Acronym 'XMHG3'
5	5	XMHRMFV	1	binary	Control block version X'03'
6	6	*	1	*	Reserved
7	7	XMHFLAG	1	binary	Flags **Bit** **Meaning When Set** **0** A data-set table was moved **1** No data-set table was moved **2** A DSNC3 table was moved **3** A DSNG3 table was moved **4-7** Reserved
8	8	XMHRETC	4	binary	Return codes **RC** **Meaning and Possible Environment** **0** Successful (XMEM and DS) **4** Time out of range (XMEM and DS) **8** Area too small (XMEM) **16** Severe error - dump call required (XMEM)
12	C	XMHLEN	4	binary	Total length of getmained sample area. If XMHRETC=8, total length needed to hold all data is returned here
12	C	XMHDSPTR	4	binary	Address of the sample area getmained by DS. Valid if XMHRETC=0 OFFSET TO FIRST SSH
16	10	XMHSSHFP	4	binary	Pointer to first SSHG3. This is an address within the requestor's address space.
20	14	XMHSSHLP	4	binary	Pointer to last SSHG3. This is an address within the requestor's address space.
24	18	XMHFRSTI	8	EBCDIC	Time of first SSH moved. Valid if XMHRETC = 0
32	20	XMHLSTTI	8	EBCDIC	Time of last SSH moved. Valid if XMHRETC = 0
40	28	XMHFRSTA	8	EBCDIC	Time of the first SSH available in the wrap around buffer
48	30	XMHLSTTA	8	EBCDIC	Time of the last SSH available in the wrap around buffer
56	38	XMHDSACI	2	binary	Index of the currently active dataset within the DSNC3 dataset names table
58	3A	*	2	*	Reserved
60	3C	XMHDSACL	8	EBCDIC	Time of the last SSH available on the active dataset

End of Programming Interface information

Chapter 7. Monitor III data reporter tables

Monitor III Table Formats

This chapter:

- Describes the data tables, and graphic parameter table used by the Monitor III data reporter
- Lists the ISPF record fields and table entries associated with creating, formatting, and displaying RMF reports

See Chapter 5, “Adding Monitor III user exits”, on page 5-1 for information on how to create user-defined reports.

Tabular report format table ERBFMTS3

Programming Interface information

The RMF format table defines the layout of RMF reports for panel display and hardcopy printing. It also ensures that each output function within RMF produces the same format.

This table contains one row for each report name and format. Each row contains information on how to edit heading and column data and contains an example for each variable name.

Variable Name	T	Variable Description	Example
FMTREPNA	K	Report name	DELAY
FMTFORMT	K	Report format identifier (not yet used)	ENGLISH
FMTRMODE	N	Report mode available (GRAPHIC/TABULAR/BOTH)	BOTH
FMTTPANL	N	Tabular report panel name	ERB3JDE
FMTTHLPP	N	Name of related help panel	ERB3JDE1
FMTLOGLN	N	Name of logical line number variable	JDEDTLLN
FMTSEQNR	N	Name of sequence number variable	JDEDTPSN
FMTCMDLN	N	Content of command line	COMMAND ===> &ZCMD ...
FMTHDR1	N	Content of header line 1 (text and variables intermixed)	... RMF DELAYS &HDRSID ..
FMTHDR2	N	Content of header line 2 (text and variables intermixed)	... Samples: &Z TIME: .
FMTSUBH1	N	Content of subheader line 1 (text and variables intermixed)	
FMTSUBH2	N	Content of subheader line 2 (text and variables intermixed)	
FMTSUBH3	N	Content of subheader line 3 (text and variables intermixed)	
FMTSUBH4	N	Content of subheader line 4 (text and variables intermixed)	
FMTSUBH5	N	Content of subheader line 5 (text and variables intermixed)	
FMTCOLH1	N	Text for column header line 1	WFL USG
FMTCOLH2	N	Text for column header line 2	NAME C DMN % % ..
FMTCOLH3	N	Text for column header line 3	
FMTHVPRE	N	Prefix used in specifying variables in header lines	&
FMTHPLCH	N	Header line placeholder replacement variable names	HDRSAMPL HDRDATE HDRTIME
FMTSPLCH	N	Subheader line placeholder replacement variable names	

Variable Name	T	Variable Description	Example
FMTCPLCH	N	Command line placeholder replacement variable names	AMT
FMTMODL1	N	Definition of model line 1 (attribute characters followed by variable names or placeholder values(Z), variable names used must be elements of the report column data table)	|JDELDAN|Z|Z |Z
FMTMODL2	N	Definition of model line 2	
FMTMODL3	N	Definition of model line 3	
FMTMATTR	N	Attribute characters used in model lines	_|¢
FMTMPLCH	N	Model line placeholder replacement variable names (ZVARS)	JDETYPE JDELDMN JDELPGN
FMTHVMAX	N	Number of variables within header lines (maximum of 20)	6
FMTSVMAX	N	Number of variables within subheader lines (maximum of 30)	0
FMTMVMAX	N	Number of variables within model lines (maximum of 30)	16
FMTCVMAX	N	Number of variables within command line (maximum of 5)	
FMTHVNnn	S	Variable name used in header lines	HDRSID
FMTHVRnn	S	Number of header line where variable is used	1
FMTHVPnn	S	Variable position within line	52
FMTHVLnn	S	Maximum variable length	15
FMTSVNxx	S	Variable name used in subheader lines	
FMTSVRxx	S	Number of subheader line where variable is used	
FMTSVPxx	S	Variable position within line	
FMTSVLxx	S	Maximum variable length	
FMTMVNyy	S	Variable name used in model lines	JDELDAN
FMTMVRyy	S	Number of model line where variable is used	1
FMTMVPyy	S	Variable position within line	2
FMTMVLyy	S	Maximum variable length	8
FMTCVNzz	S	Variable name used in command line	ZCMD
FMTCVPzz	S	Variable position within line	14
FMTCVLzz	S	Maximum variable length	51

Report format

Note:

K - KEY type variable

N - NAMES type variable

S - EXTENSION type variable

nn = unique number for each variable used in the header lines

xx = unique number for each variable used in the subheader lines

yy = unique number for each variable used in the model lines

zz = unique number for each variable used in the command line

End of Programming Interface information

Header data table ERBHDRS3

Programming Interface information

The RMF header data table provides the variable heading information in one table row for each report.

Variable Name	T	Variable Description	Example
HDRREPNA	K	Report name	DELAY
ERBSID	N	System identifier	AQXA
ERBHCTXT	N	Hardcopy text constant	HARDCOPY
ERBSAMPL	N	Sample count	100
ERBDATE	N	Starting date	07/02/02
ERBTIME	N	Starting time	10.35.00
ERBRANGE	N	Time range value	100
ERBRMFVD	N	RMF version	RMF V1R2
ERBSPXID	N	Sysplex ID	RMFPLEX
ERBSNUM	N	Number of systems within sysplex	5
ERBSAMWL	N	Number of WLM samples	100
	S	The variable data for subheader lines has to be kept in extension values of this table. Example for STORR report.	

Note:

K - KEY type variable

N - NAMES type variable

S - EXTENSION type variable

End of Programming Interface information

Report data tables

Programming Interface information

Each of the following report data tables indicates in column **Report** whether a value is part of the Monitor III report (Yes), is part of a pop-up window (Pop-Up), or is available through the Monitor III Utility (Util).

Column **T** indicates whether it is a KEY-type variable (K) or a NAMES-type variable (N).

CACHDET - Tabular report data table ERBCADT3

RMF builds ERBCADT3 when using CACHDET as a report type.

Name	T	Description of the Variable	Report
CADDTLLN	K	Logical line number	-
CADDTPSN	K	Sequence number	-
CADPVOLU	N	Volume	Yes
CADPDEVN	N	Device Number	Yes
CADPSSID	N	SSID	Yes
CADPIOP	N	I/O percentage	Yes
CADPIO	N	I/O rate	Yes
CADPHITP	N	Hit percentage	Yes
CADPREAD	N	Cache hit rate READ	Yes
CADPDFW	N	Cache hit rate DFW	Yes
CADPCFW	N	Cache hit rate CFW	Yes
CADPTOT	N	DASD I/O rate total	Yes
CADPSTAG	N	DASD I/O rate stage	Yes
CADPSEQ	N	Sequential rate	Yes
CADPASYN	N	Async rate	Yes
CADICACH	N	Cache state	Yes
CADIDFW	N	DFW state	Pop-Up
CADIPIN	N	Pinned state	Pop-Up
CADNRRA	N	Norm Read rate	Pop-Up
CADNRHI	N	Norm Read hit rate	Pop-Up
CADNRHIP	N	Norm Read hit percentage	Pop-Up
CADNWRA	N	Norm Write rate	Pop-Up
CADNWFA	N	Norm Write fast rate	Pop-Up
CADNWHI	N	Norm Write hit rate	Pop-Up
CADNWHIP	N	Norm Write hit percentage	Pop-Up
CADNREAP	N	Norm Read percentage	Pop-Up
CADNTRA	N	Norm Tracks rate	Pop-Up
CADSRRA	N	Seq Read rate	Pop-Up
CADSRHI	N	Seq Read hit rate	Pop-Up
CADSRHIP	N	Seq Read hit percentage	Pop-Up

CACHDET data

Name	T	Description of the Variable	Report
CADSWRA	N	Seq Write rate	Pop-Up
CADSWFA	N	Seq Write fast rate	Pop-Up
CADSWHI	N	Seq Write hit rate	Pop-Up
CADSWHIP	N	Seq Write hit percentage	Pop-Up
CADSREAP	N	Seq Read percentage	Pop-Up
CADSTRA	N	Seq Tracks rate	Pop-Up
CADCRRA	N	CFW Read rate	Pop-Up
CADCRHI	N	CFW Read hit rate	Pop-Up
CADCRHIP	N	CFW Read hit percentage	Pop-Up
CADCWRA	N	CFW Write rate	Pop-Up
CADCWHI	N	CFW Write hit rate	Pop-Up
CADCWHIP	N	CFW Write hit percentage	Pop-Up
CADCREAP	N	CFW Read percentage	Pop-Up
CADTRRA	N	Total Read rate	Pop-Up
CADTRHI	N	Total Read hit rate	Pop-Up
CADTRHIP	N	Total Read hit percentage	Pop-Up
CADTWRA	N	Total Write rate	Pop-Up
CADTWFA	N	Total Write fast rate	Pop-Up
CADTWHI	N	Total Write hit rate	Pop-Up
CADTWHIP	N	Total Write hit percentage	Pop-Up
CADTREAP	N	Total Read percentage	Pop-Up
CADMDFWB	N	DFW bypass	Pop-Up
CADMNICL	N	Non-cache ICL	Pop-Up
CADMCWRI	N	CKD write	Pop-Up
CADMRCRM	N	Read miss	Pop-Up
CADMCFWB	N	CFW bypass	Pop-Up
CADMNBYP	N	Non-cache bypass	Pop-Up
CADMCHIT	N	CKD hits	Pop-Up
CADMRCWP	N	Write prom	Pop-Up
CADMDFWI	N	DFW inhibit	Pop-Up

CACHSUM - Tabular report data table ERBCAST3

RMF builds ERBCAST3 when using CACHSUM as a report type.

Name	T	Description of the Variable	Report
CASDTLLN	K	Logical line number	-
CASDTPSN	K	Sequence number	-
CASPSSID	N	SSID	Yes
CASPCUID	N	CUID	Yes
CASPTYPM	N	Type-Mod	Yes
CASPSIZE	N	Storage size	Yes

Name	T	Description of the Variable	Report
CASPIO	N	I/O rate	Yes
CASPHITP	N	Hit percentage	Yes
CASPHIT	N	Hit rate	Yes
CASPMTOT	N	Miss total rate	Yes
CASPMSTG	N	Miss stage rate	Yes
CASPREAP	N	Read percentage	Yes
CASPSEQ	N	Sequential rate	Yes
CASPASYN	N	Async rate	Yes
CASPOFF	N	Off rate	Yes
CASNRRA	N	Norm Read rate	Pop-Up
CASNRHI	N	Norm Read hit rate	Pop-Up
CASNRHIP	N	Norm Read hit percentage	Pop-Up
CASNWRA	N	Norm Write rate	Pop-Up
CASNWFA	N	Norm Write fast rate	Pop-Up
CASNWHI	N	Norm Write hit rate	Pop-Up
CASNWHIP	N	Norm Write hit percentage	Pop-Up
CASNREAP	N	Norm Read percentage	Pop-Up
CASNTRA	N	Norm Tracks rate	Pop-Up
CASSRRA	N	Seq Read rate	Pop-Up
CASSRHI	N	Seq Read hit rate	Pop-Up
CASSRHIP	N	Seq Read hit percentage	Pop-Up
CASSWRA	N	Seq Write rate	Pop-Up
CASSWFA	N	Seq Write fast rate	Pop-Up
CASSWHI	N	Seq Write hit rate	Pop-Up
CASSWHIP	N	Seq Write hit percentage	Pop-Up
CASSREAP	N	Seq Read percentage	Pop-Up
CASSTRA	N	Seq Tracks rate	Pop-Up
CASCRRA	N	CFW Read rate	Pop-Up
CASCRHI	N	CFW Read hit rate	Pop-Up
CASCRHIP	N	CFW Read hit percentage	Pop-Up
CASCWRA	N	CFW Write rate	Pop-Up
CASCWHI	N	CFW Write hit rate	Pop-Up
CASCWHIP	N	CFW Write hit percentage	Pop-Up
CASCREAP	N	CFW Read percentage	Pop-Up
CASTRRA	N	Total Read rate	Pop-Up
CASTRHI	N	Total Read hit rate	Pop-Up
CASTRHIP	N	Total Read hit percentage	Pop-Up
CASTWRA	N	Total Write rate	Pop-Up
CASTWFA	N	Total Write fast rate	Pop-Up
CASTWHI	N	Total Write hit rate	Pop-Up
CASTWHIP	N	Total Write hit percentage	Pop-Up

CACHSUM data

Name	T	Description of the Variable	Report
CASTREAP	N	Total Read percentage	Pop-Up
CASMCACH	N	Cache state	Pop-Up
CASMCCON	N	Cache configured	Pop-Up
CASMCAVL	N	Cache available	Pop-Up
CASMCOFF	N	Cache offline	Pop-Up
CASMCPIN	N	Cache pinned	Pop-Up
CASMNVS	N	NVS state	Pop-Up
CASMNCON	N	NVS configured	Pop-Up
CASMNPIN	N	NVS pinned	Pop-Up

CFACT - Tabular report data table ERBCFAT3

RMF builds ERBCFAT3 when using CFACT as a report type.

Name	T	Description of the Variable	Report
CFADTLLN	K	Logical line number	-
CFADTPSN	K	Sequence number	-
CFAPSTRU	N	Structure name	Yes
CFAPTYPE	N	Structure type	Yes
CFAPSTAT	N	Structure status	Yes
CFAPSYS	N	System name	Yes
CFAPSYNR	N	Sync rate	Yes
CFAPASS	N	Sync average service time	Yes
CFAPASYR	N	Async rate	Yes
CFAPAAS	N	Async average service time	Yes
CFAPACHG	N	Async changed %	Yes
CFAPADEL	N	Async delay %	Yes
CFAPUTD2	N	Structure information	Yes
CFAINAM	N	Coupling facility name	Yes
CFAISTRU	N	Structure name	Pop-Up
CFAITYPE	N	Structure type	Pop-Up
CFAICNAM	N	Connection name	Pop-Up
CFAICJOB	N	Job name	Pop-Up
CFAICSTA	N	Status	Pop-Up
CFAICASI	N	ASID	Pop-Up
CFAICLVL	N	CF level	Pop-Up
CFAISTRS	N	Structure size	Pop-Up
CFAILEL	N	List entries total (LIST/LOCK only)	Pop-Up
CFAILEM	N	List entries current (LIST/LOCK only)	Pop-Up
CFAIMAE	N	Data elements total (LIST only)	Pop-Up
CFAICUE	N	Data elements current (LIST only)	Pop-Up
CFAILTL	N	Lock entries total (LIST/LOCK only)	Pop-Up

Name	T	Description of the Variable	Report
CFAILTM	N	Lock entries current (LIST/LOCK only)	Pop-Up
CFAIDEN	N	Directory entries total (CACHE only)	Pop-Up
CFAIDEC	N	Directory entries current (CACHE only)	Pop-Up
CFAIDEL	N	Data elements total (CACHE only)	Pop-Up
CFAIDAC	N	Data elements current (CACHE only)	Pop-Up
CFAICONT	N	Contention %	Pop-Up
CFAIFCON	N	False Contention % (LOCK only)	Pop-Up
CFAIREQR	N	Request rate (CACHE only)	Pop-Up
CFAIREAR	N	Read rate (CACHE only)	Pop-Up
CFAIWRIR	N	Write rate (CACHE only)	Pop-Up
CFAICAOR	N	Castout rate (CACHE only)	Pop-Up
CFAIXIR	N	XI rate (CACHE only)	Pop-Up
CFAIDER	N	Directory reclaims (CACHE only)	Pop-Up

CFOVER - Tabular report data table ERBCFOT3

RMF builds ERBCFOT3 when using CFOVER as a report type.

Name	T	Description of the Variable	Report
CFODTLLN	K	Logical line number	-
CFODTPSN	K	Sequence number	-
CFOPNAM	N	Coupling facility name	Yes
CFOPMOD	N	Model	Yes
CFOPVER	N	Version	Yes
CFOPLVL	N	CF level	Yes
CFOPUTIP	N	Processor utilization %	Yes
CFOPDEF	N	Processor defined	Yes
CFOPEFF	N	Processor effective	Yes
CFOPREQR	N	Request rate	Yes
CFOPTSD	N	Storage size	Yes
CFOPTSF	N	Storage available	Yes

CFSYS - Tabular report data table ERBCFST3

RMF builds ERBCFST3 when using CFSYS as a report type.

Name	T	Description of the Variable	Report
CFSDTLLN	K	Logical line number	-
CFSDTPSN	K	Sequence number	-
CFSPNAM	N	Coupling facility name	Yes
CFSPSYS	N	System name	Yes
CFSPSDEL	N	Subchannel delay %	Yes
CFSPPTHA	N	Paths available	Yes

CFSYS data

Name	T	Description of the Variable	Report
CFSPPDEL	N	Paths delay %	Yes
CFSPSYNR	N	Sync rate	Yes
CFSPASS	N	Sync average service time	Yes
CFSPASYR	N	Async rate	Yes
CFSPAAS	N	Async average service time	Yes
CFSPACHG	N	Async changed %	Yes
CFSPADEL	N	Async delay %	Yes
CFSINAM	N	Coupling facility name	Pop-Up
CFSISCG	N	Subchannels generated	Pop-Up
CFSISCU	N	Subchannels in use	Pop-Up
CFSISCL	N	Subchannels max	Pop-Up
CFSIPATH	N	Paths Ids	Pop-Up

CHANNEL - Tabular report data table ERBCHAT3

RMF builds ERBCHAT3 when using CHANNEL as a report type.

Name	T	Description of the Variable	Report
CHADTLLN	K	Logical line number	-
CHADTPSN	K	Sequence number	-
CHACPIVC	N	Channel path ID	Yes
CHACPNVC	N	Number of DCM-managed channels	Yes
CHACGVC	N	Channel type generation	Yes
CHACPTVC	N	Channel path type	Yes
CHACSIVC	N	Channel shared indication	Yes
CHACPUVC	N	Partition utilization percent	Yes
CHACTUVC	N	Total utilization percent	Yes
CHACTBVC	N	Bus utilization percent	Yes
CHACPRVC	N	Partition transfer rate (Read) in B/sec	Yes
CHACTRVC	N	Total transfer rate (Read) in B/sec	Yes
CHACPWVC	N	Partition transfer rate (Write) in B/sec	Yes
CHACTWVC	N	Total transfer rate (Write) in B/sec	Yes
CHACPMVC	N	Partition message sent rate	Yes
CHACTMVC	N	Total message sent rate	Util
CHACPSVC	N	Partition message sent size	Yes
CHACTSVC	N	Total message sent size	Util
CHACSFVC	N	Partition message sent fail rate	Yes
CHACPFVC	N	Partition receive fail rate	Yes
CHACTFVC	N	Total receive fail rate	Util

CPC - Tabular report data table ERBCPCT3

RMF builds ERBCPCT3 when using CPC as a report type.

Name	T	Description of the Variable	Report
CPCDTLLN	K	Logical line number	-
CPCDTPSN	K	Sequence number	-
CPCPPNAM	N	Partition name	Yes
CPCPDMSU	N	Defined capacity limit	Yes
CPCPAMSU	N	Actual consumed MSUs	Yes
CPCPCAPI	N	Defined capping (yes/no/mix)	Yes
CPCPLPNO	N	Average number of logical processors	Yes
CPCPLEFU	N	Logical processor effective utilization %	Yes
CPCPLTOU	N	Logical processor total utilization %	Yes
CPCPPLMU	N	Physical LPAR utilization %	Yes
CPCPPEFU	N	Physical processor effective utilization %	Yes
CPCPPTOU	N	Physical processor total utilization %	Yes
CPCPIND	N	Type/partition indicator	No

DELAY - Tabular report data table ERBJDET3

RMF builds ERBJDET3 when using DELAY as a report type.

Name	T	Description of the Variable	Report
JDEDTLLN	K	Logical line number	-
JDEDTPSN	K	Sequence number	-
JDELDAN	N	Jobname or summary	Yes
JDETYPE	N	Class (A, B, E, O, S, or T)	Util
JDETYPX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
JDELDMN	N	Domain	Yes
JDELPGN	N	Performance group	Yes
JDEPSVCL	N	Service class name	Yes
JDEGMIP	N	GMI removal	Yes
JDELWFL	N	Work flow percentage	Yes
JDELUSG	N	Using percentage	Yes
JDELDEL	N	Delay percentage	Yes
JDELIDL	N	Idle percentage	Yes
JDELUKN	N	Unknown percentage	Yes
JDELPROC	N	Processor delay percentage	Yes
JDELDEV	N	Device delay percentage	Yes
JDELSTOR	N	Storage delay percentage	Yes
JDELSUBS	N	JES, HSM, and XCF delay percentage	Yes
JDELOPER	N	Operator delay percentage	Yes
JDELENQ	N	ENQ delay percentage	Yes

DELAY data

Name	T	Description of the Variable	Report
JDELJES	N	JES delay percentage	Util
JDELHSM	N	HSM delay percentage	Util
JDELXCF	N	XCF delay percentage	Util
JDELMNT	N	Operator mount delay percentage	Util
JDELMES	N	Operator message delay percentage	Util
JDELQUI	N	Operator quiesce delay percentage	Util
JDELREAS	N	Primary reason	Yes

DEV - Tabular report data table ERBDEVT3

RMF builds table ERBDEVT3 when using DEV as a report type.

Name	T	Description of the Variable	Report
DEVDTLLN	K	Logical line number	-
DEVDTPSN	K	Sequence number	-
DEVPJOB	N	Jobname	Yes
DEVPCLA	N	Class (A, B, O, S, or T)	Yes
DEVPDMN	N	Domain	Yes
DEVPPGN	N	Performance group	Yes
DEVPSVCL	N	Service class name	Yes
DEVPODEL	N	Overall delay percentage	Yes
DEVPOUSE	N	Overall using percentage	Yes
DEVPCON	N	Connect time	Yes
DEV1SDEL	N	Delay percentage causes by volser1	Yes
DEV1VOLU	N	Volume serial number volser1	Yes
DEV2SDEL	N	Delay percentage caused by volser2	Yes
DEV2VOLU	N	Volume serial number volser2	Yes
DEV3SDEL	N	Delay percentage cause by volser3	Yes
DEV3VOLU	N	Volume serial number volser3	Yes
DEV4SDEL	N	Delay percentage caused by volser4	Yes
DEV4VOLU	N	Volume serial number volser4	Yes

DEVR - Tabular report data table ERBDVRT3

RMF builds ERBDVRT3 when using DEVR as a report type.

Name	T	Description of the Variable	Report
DVRDTLLN	K	Logical line number	-
DVRDTPSN	K	Sequence number	-
DVRPVOLU	N	Volser	Yes
DVRPDEVN	N	Device number	Yes
DVRPIDEN	N	Device indication (model)	Yes
DVRPSTAT	N	Status	Yes

Name	T	Description of the Variable	Report
DVRPEXP	N	Number of exposures	Yes
DVRPACTV	N	Percentage of active time	Yes
DVRPCONN	N	Percentage of connect time	Yes
DVRPDISC	N	Percentage of disconnect time	Yes
DVRPPEND	N	Percentage of pending time	Util
DVRPDLYR	N	Pending delay reason header	Yes
DVRPDLYP	N	Pending delay reason percentage	Yes
DVRACTRT	N	Device activity rate	Yes
DVRRESPT	N	Response Time	Yes
DVRIOSQT	N	IOS queue time	Util
DVRPDVBT	N	Percentage of device busy delay time	Util
DVRPCUBT	N	Percentage of control unit busy delay time	Util
DVRPSPBT	N	Percentage of director port busy delay time	Util
DVRPJOBN	N	Jobname	Yes
DVRPCLA	N	Class (A, B, O, S, or T)	Yes
DVRPDMN	N	Domain	Yes
DVRPPGN	N	Performance group number	Yes
DVRPSUSE	N	Percentage of using	Yes
DVRPSDEL	N	Percentage of delay	Yes
DVRPSVCL	N	Service class	Yes
DVRPKIND	N	Device type indicator	Util
DVRPLCUN	N	Logical control unit ID	Util

DI - Tabular report data table ERBDSIT3

RMF builds ERBDSIT3 when using DI as a report type.

Name	T	Description of the Variable	Report
DSIDTLLN	K	Logical line number	-
DSIDTPSN	K	Sequence number	-
DSI1SID	N	System identifier	Yes
DSI1DATE	N	Starting date	Yes
DSI1TIME	N	Starting time	Yes
DSI1DDNM	N	DD-name	Yes
DSI1DSNM	N	Data set name	Yes
DSI2DATE	N	Ending date	Yes
DSI2TIME	N	Ending time	Yes
DSI2MESS	N	Message field	Yes

DSND - Tabular report data table ERBDNDT3

RMF builds ERBDNDT3 when using DSND as a report type.

Name	T	Description of the Variable	Report
DNDDTLLN	K	Logical line number	-
DNDDTPSN	K	Sequence number	-
DNDPDSN	N	Data set name	Yes
DNDPVOLU	N	Volume serial	Yes
DNDPJOBN	N	Jobname	Yes
DNDPASID	N	ASID	Yes
DNDPDUSG	N	DUSG (Using %)	Yes
DNDPDDLY	N	DDLY (Dealy %)	Yes

DSNJ - Tabular report data table ERBDNJT3

RMF builds ERBDNJT3 when using DSNJ as a report type.

Name	T	Description of the Variable	Report
DNJDTLLN	K	Logical line number	-
DNJDTPSN	K	Sequence number	-
DNJPASID	N	ASID	Yes
DNJPDSN	N	Data set name	Yes
DNJPVOLU	N	Volume	Yes
DNJPDEVN	N	Device number	Yes
DNJPDUSG	N	DUSG (Using %)	Yes
DNJPDDLY	N	DDLY (Delay %)	Yes

DSNV - Tabular report data table ERBDNVT3

RMF builds ERBDNVT3 when using DSNV as a report type.

Name	T	Description of the Variable	Report
DNVDTLLN	K	Logical line number	-
DNVDTPSN	K	Sequence number	-
DNVPDSN	N	Data set name	Yes
DNVPJOBN	N	Jobname	Yes
DNVPASID	N	ASID	Yes
DNVPDUSG	N	DUSG (Using %)	Yes
DNVPDDLY	N	DDLY (Delay %)	Yes

ENCLAVE - Tabular report data table ERBENCT3

RMF builds ERBENCT3 when using ENCLAVE as a report type.

Name	T	Description of the Variable	Report
ENCDTLLN	K	Logical line number	-

Name	T	Description of the Variable	Report
ENCDTPSN	K	Sequence number	-
ENCENAME	N	Enclave name (generated)	Yes
ENCCLASS	N	Service class / performance group	Yes
ENCGOAL	N	Goal time (in goal mode)	Yes
ENCGPERC	N	Goal percent (in goal mode)	Yes
ENCPER	N	Period	Yes
ENCDENC	N	Dependent enclave indicator	Yes
ENCXENC	N	Multi-system enclave indicator	Yes
ENCTCPU	N	Total CPU time (seconds)	Yes
ENCDCPU	N	Delta CPU time (seconds)	Pop-Up
ENCDCPUP	N	Delta CPU percentage in Monitor III range	Yes
ENCSAMP	N	Total execution samples	Pop-Up
ENCTUSG	N	% Total using samples	Yes
ENCTDLY	N	% Total delay samples	Yes
ENCIDLE	N	% Idle	Yes
ENCCUSG	N	% CPU using	Pop-Up
ENCCDLY	N	% CPU delay	Pop-Up
ENCIUSG	N	% I/O using	Pop-Up
ENCIDLY	N	% I/O delay	Pop-Up
ENCCCAP	N	% CPU capping	Pop-Up
ENCSTOR	N	% Storage delay	Pop-Up
ENCUNKN	N	% Unknown	Pop-Up
ENCQUED	N	% Queue delay	Pop-Up
ENCESTYP	N	Subsystem type	Pop-Up
ENCEOWNM	N	Owner name	Pop-Up
ENCEOSYS	N	Owner system	Pop-Up
ENCXTOKN	N	Export token	Pop-Up
ENCATTN	N	Number of attributes in table	Pop-Up
ENCATT01	N	Attribute 01	Pop-Up
ENCATT02	N	Attribute 02	Pop-Up
ENCATT03	N	Attribute 03	Pop-Up
ENCATT04	N	Attribute 04	Pop-Up
ENCATT05	N	Attribute 05	Pop-Up
ENCATT06	N	Attribute 06	Pop-Up
ENCATT07	N	Attribute 07	Pop-Up
ENCATT08	N	Attribute 08	Pop-Up
ENCATT09	N	Attribute 09	Pop-Up
ENCATT10	N	Attribute 10	Pop-Up
ENCATT11	N	Attribute 11	Pop-Up
ENCATT12	N	Attribute 12	Pop-Up
ENCATT13	N	Attribute 13	Pop-Up

ENCLAVE data

Name	T	Description of the Variable	Report
ENCATT14	N	Attribute 14	Pop-Up
ENCATT15	N	Attribute 15	Pop-Up
ENCATT16	N	Attribute 16	Pop-Up

ENQ - Tabular report data table ERBENQT3

RMF builds ERBENQT3 when using ENQ as a report type.

Name	T	Description of the Variable	Report
ENQDTLLN	K	Logical line number	-
ENQDTPSN	K	Sequence number	-
ENQPWJOB	N	Jobname of waiting job	Yes
ENQPODEL	N	Overall delay percentage	Yes
ENQPRDEL	N	Percentage of delay for the resource	Yes
ENQPWSTT	N	Status of waiting job	Yes
ENQPMAJS	N	Resource major name and scope or minor name	Yes
ENQPHDEL	N	Holding percentage for the holding job	Yes
ENQPHJOB	N	Jobname of holding job or system name for holding job	Yes
ENQPHSTT	N	Status for the holding job	Yes

ENQR - Tabular report data table ERBEQRT3

RMF builds ERBEQRT3 when using ENQR as a report type.

Name	T	Description of the Variable	Report
EQRDTLLN	K	Logical line number	-
EQRDTPSN	K	Sequence number	-
EQRPMAJS	N	Resource major name and scope or resource minor name	Yes
EQRPRDEP	N	Percentage of delay for the delayed job	Yes
EQRPWJOB	N	Jobname of delayed job	Yes
EQRPWSTT	N	Status of delayed job	Yes
EQRPHDEP	N	Holding percentage for the holding job	Yes
EQRPHJOB	N	Jobname of holding job or system name	Yes
EQRPHSTT	N	Status of holding job	Yes

HSM - Tabular report data table ERBHSMT3

RMF builds ERBHSMT3 when using HSM as a report type. The table variables are identical to the variables of the ERBJEST3 table; see the ERBJEST3 table for more information.

IOQUEUE - Tabular report data table ERBIOQT3

RMF builds ERBIOQT3 when using IOQUEUE as a report type.

Name	T	Description of the Variable	Report
IOQDTLLN	K	Logical line number	-
IOQDTPSN	K	Sequence number	-
IOQCPIVC	N	Channel path ID	Yes
IOQDCMVC	N	DCM-managed channels	Yes
IOQPCUVC	N	Physical CU string	Yes
IOQMMNVC	N	Minimum number of DCM-managed channels used	Yes
IOQMMXVC	N	Maximum number of DCM-managed channels used	Yes
IOQMDFVC	N	Defined number of DCM-managed channels	Yes
IOQLCUVC	N	Logical control unit ID	Yes
IOQCRTVC	N	Contention rate	Yes
IOQDQLVC	N	Delay queue length	Yes
IOQCPTVC	N	Channel path ID taken	Yes
IOQSPBVC	N	Director port busy percent	Yes
IOQCUBVC	N	Control unit busy percent	Yes

JES - Tabular report data table ERBJEST3

RMF builds ERBJEST3 when using JES as a report type.

Name	T	Description of the Variable	Report
HJSDTLLN	K	Logical line number	-
HJSDTPSN	K	Sequence number	-
HJSPJOB	N	Jobname	Yes
HJSPODEL	N	Overall delay percentage	Yes
HJS1FDEL	N	Delay percentage	Yes
HJS1FCNR	N	Function code	Yes
HJS1EXPL	N	Explanation	Yes
HJS2FDEL	N	Delay percentage	Util
HJS2FCNR	N	Function code	Util
HJS2EXPL	N	Explanation	Util

JOB - Tabular report data table ERBJDJT3

RMF builds ERBJDJT3 when using JOB as a report type.

Name	T	Description of the Variable	Report
JDJDTLLN	K	Logical line number	-
JDJDTPSN	K	Sequence number	-
JDJLDAN	N	Jobname or summary	Yes
JDJLASID	N	Address space identification	Yes

JOB data

Name	T	Description of the Variable	Report
JDJCLASS	N	Class (A, B, E, O, S, or T)	Util
JDJCLASX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
JDJLDMN	N	Domain	Yes
JDJLPGN	N	Performance group	Yes
JDJPSVCL	N	Service class name	Yes
JDJLWFL	N	Work flow percentage	Yes
JDJLUSP	N	Processor using percentage	Yes
JDJLUSD	N	Device using percentage	Yes
JDJLUSG	N	Using percentage	Util
JDJLDEL	N	Delay percentage	Yes
JDJLIDL	N	Idle percentage	Yes
JDJLUKN	N	Unknown percentage	Yes
JDJLPROC	N	Processor delay percentage	Yes
JDJLDEV	N	Device delay percentage	Yes
JDJLSTOR	N	Storage delay percentage	Yes
JDJLSUBS	N	SUBS delay percentage	Yes
JDJLOPER	N	Operator delay percentage	Yes
JDJLENQ	N	ENQ delay percentage	Yes
JDJLJES	N	JES delay percentage	Util
JDJLHSM	N	HSM delay percentage	Util
JDJLXCF	N	XCF delay percentage	Util
JDJLMNT	N	Operator mount delay percentage	Util
JDJLMES	N	Operator message delay percentage	Util
JDJLQUI	N	Operator quiesce delay percentage	Util
JDJLREAS	N	Primary reason	Yes

OPD - Tabular report data table ERBOPDT3

RMF builds ERBOPDT3 when using OPD as a report type.

Name	T	Description of the Variable	Report
OPDDTLLN	K	Logical line number	-
OPDDTPSN	K	Sequence number	-
OPDPJOBN	N	Jobname	Yes
OPDPUSEN	N	User name	Yes
OPDPASID	N	ASID	Yes
OPDPASIX	N	Hexadecimal ASID	Yes
OPDPPRID	N	Process ID	Yes
OPDPPPID	N	Parent's process ID	Yes
OPDPLATW	N	Waiting for process latch	Yes
OPDPSTAT	N	Process state	Yes

Name	T	Description of the Variable	Report
OPDPAPPL	N	Percentage of TCB and SRB time	Yes
OPDPTOT	N	Total computing time since process has been started	Yes
OPDPSERV	N	Server type	Yes
OPDIPRID	N	Process ID	Pop-up
OPDIPPID	N	Parent’s process ID	Pop-up
OPDIJOBN	N	Jobname	Pop-up
OPDIUSEN	N	User name	Pop-up
OPDIASID	N	ASID	Pop-up
OPDIASIX	N	Hexadecimal ASID	Pop-up
OPDITIMD	N	Start time/date	Pop-up
OPDIAPPL	N	Percentage of TCB and SRB time	Pop-up
OPDITOT	N	Total computing time since process has been started	Pop-up
OPDILPID	N	Latch process ID the process is waiting for (0 = not waiting)	Pop-up
OPDICMD	N	Command buffer	Pop-up
OPDISERN	N	Server name	Pop-up
OPDISERV	N	Server type	Pop-up
OPDIACTF	N	Number of active files	Pop-up
OPDIMAXF	N	Maximum files	Pop-up
OPDISTAT	N	Process state	Pop-up
OPDISTA1	N	1. possible state	Pop-up
OPDISTA2	N	2. possible state	Pop-up
OPDISTA3	N	3. possible state	Pop-up

PROC - Tabular report data table ERBPRCT3

RMF builds ERBPRCT3 when using PROC as a report type.

Name	T	Description of the Variable	Report
PRCDTLLN	K	Logical line number	-
PRCDTPSN	K	Sequence number	-
PRCPJOB	N	Jobname	Yes
PRCPCLA	N	Class (A, B, E, O, S, or T)	Util
PRCPCLAX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
PRCPDMN	N	Domain number	Yes
PRCPPGN	N	Performance group number	Yes
PRCPODEL	N	Overall delay percentage	Yes
PRCPOUSE	N	Using percentage	Yes
PRCPTST	N	Overall application percentage	Yes
PRCPVEC	N	Vector time ratio	Util

PROC data

Name	T	Description of the Variable	Report
PRCPSVCL	N	Service class name	Yes
PRCPCAPD	N	Capping delay percentage (goal mode)	Util
PRCPETST	N	EAppl percentage	Yes
PRCPTCBT	N	TCB percentage	Util
PRCPSRBT	N	SRB percentage	Util
PRCPPCST	N	Preemptable or client SRB percentage	Util
PRCPEPST	N	Preemptable or client SRB and enclave percentage	Util
PRC1SDEL	N	Delay percentage caused by jobname1	Yes
PRC1JOBN	N	Jobname1	Yes
PRC2SDEL	N	Delay percentage caused by jobname2	Yes
PRC2JOBN	N	Jobname2	Yes
PRC3SDEL	N	Delay percentage caused by jobname3	Yes
PRC3JOBN	N	Jobname3	Yes

RLSDS - Tabular report data table ERBVRDT3

RMF builds ERBVRDT3 when using RLSDS as a report type.

Name	T	Description of the Variable	Report
VRDDTLLN	K	Logical line number	-
VRDDTPSN	K	Sequence number	-
VRDPNAM	N	VSAM sphere name, Data set name, MVS system name, Access type, Response time, Read rate, Read BMF hit percentage, Read CF hit percentage	Yes
VRDPRDAS	N	Read DASD hit percentage	Yes
VRDPBMFV	N	BMF valid percentage	Yes
VRDPBMFF	N	BMF false invalid percentage	Yes
VRDPWRTE	N	Write rate	Yes
VRDPCALO	N	Castout lock percentage	Util
VRDPREDA	N	Redo activity percentage	Util
VRDPRRED	N	Recursive redo percentage	Util
VRDPIND	N	Report indication	Util

RLSLRU - Tabular report data table ERBVRLT3

RMF builds ERBVRLT3 when using RLSLRU as a report type.

Name	T	Description of the Variable	Report
VRLDTLLN	K	Logical line number	-
VRLDTPSN	K	Sequence number	-
VRLPSYS	N	MVS system name	Yes
VRLPACT	N	Average CPU time	Yes
VRLPBSG	N	Buffer size goal	Yes
VRLPBSH	N	Buffer size high	Yes

Name	T	Description of the Variable	Report
VRLPBSO	N	Buffer percentage accelerated	Yes
VRLPBSS	N	Buffer percentage reclaiming	Yes
VRLPABMF	N	Average BMF hit percentage	Yes
VRLPACAC	N	Average Cache hit percentage	Yes
VRLPADAS	N	Average DASD hit percentage	Yes
VRLPCALO	N	Castout lock percentage	Util
VRLPREDA	N	Redo activity percentage	Util
VRLPRRED	N	Recursive redo percentage	Util
VRLISYS	N	MVS system name	Pop-Up
VRLILSn	N	Buffer size low value nK where n is 2, 4, ..., 32	Pop-Up
VRLIHSn	N	Buffer size high value nK where n is 2, 4, ..., 32	Pop-Up
VRLICSn	N	Buffer size average value nK where n is 2, 4, ..., 32	Pop-Up

RLSSC - Tabular report data table ERBVRST3

RMF builds ERBVRST3 when using RLSSC as a report type.

Name	T	Description of the Variable	Report
VRSDTLLN	K	Logical line number	-
VRSDTPSN	K	Sequence number	-
VRSPNAM	N	Storage class name, MVS system name, CF structure name, Access type	Yes
VRSPRTIM	N	Average response time	Yes
VRSPRRTE	N	Read rate	Yes
VRSPRBMF	N	Read BMF hit percentage	Yes
VRSPRCF	N	Read CF hit percentage	Yes
VRSPRDAS	N	Read DASD hit percentage	Yes
VRSPBMFV	N	BMF valid percentage	Yes
VRSPBMFF	N	BMF false invalid percentage	Yes
VRSPWRTE	N	Write rate	Yes
VRSPIND	N	Report indication	Util

STOR - Tabular report data table ERBSTRT3

RMF builds ERBSTRT3 when using STOR as a report type.

Name	T	Description of the Variable	Report
STRDTLLN	K	Logical line number	-
STRDTPSN	K	Sequence number	-
STRPJOB	N	Jobname	Yes
STRPCLA	N	Class (A, B, O, S, or T)	Yes
STRPDMN	N	Domain number	Yes
STRPPGN	N	Performance group number	Yes
STRPSVCL	N	Service class name	Yes

STOR data

Name	T	Description of the Variable	Report
STRPODEL	N	Overall delay percentage	Yes
STR1SDEL	N	Delay percentage COMM	Yes
STR2SDEL	N	Delay percentage local	Yes
STR3SDEL	N	Delay percentage VIO	Util
STR4SDEL	N	Delay percentage SWAP	Yes
STR5SDEL	N	Delay percentage OUTR	Yes
STR6SDEL	N	Cross memory delay %	Util
STR7SDEL	N	Hiperspace delay %	Util
STR8SDEL	N	Other delays % (including VIO, XMEM and HIPR)	Yes
STRPACTV	N	Average ACTV frames	Util
STRPFIXD	N	Average fixed frames total	Util
STRPIDLE	N	Average IDLE frames	Util
STRPWSET	N	Average working set frames	Yes
STRPWSEX	N	Average ES working set frames	Yes

STORC - Tabular report data table ERBCSUT3

RMF builds ERBCSUT3 when using STORC as a report type.

Name	T	Description of the Variable	Report
CSUDTLLN	K	Logical line number	-
CSUDTPSN	K	Sequence number	-
CSXNAME	N	Jobname	Yes
CSXACT	N	Active column	Yes
CSXCLA	N	Class (A, B, O, S, or T)	Yes
CSXDMN	N	Domain number	Yes
CSXPGN	N	Performance group number	Yes
CSXCSN	N	Service class name	Yes
CSXASID	N	Address space identifier	Yes
CSXTIME	N	Elapsed time	Yes
CSXPCSA	N	Percentage of CSA	Yes
CSXPECS	N	Percentage of ECSA	Yes
CSXPSQA	N	Percentage of SQA	Yes
CSXPESQ	N	Percentage of ESQA	Yes
CSXACSA	N	Amount of CSA	Yes
CSXAECS	N	Amount of ECSA	Yes
CSXASQA	N	Amount of SQA	Yes
CSXAESQ	N	Amount of ESQA	Yes
CSXJESID	N	JES identifier	Util
CSXTDATE	N	Termination date	Util
CSXTTIME	N	Termination time	Util

STORCR - Tabular report data table ERBCRST3

RMF builds ERBCRST3 when using STORCR as a report type.

Name	T	Description of the Variable	Report
CSUDTLLN	K	Logical line number	-
CSUDTPSN	K	Sequence number	-
CSXNAME	N	Jobname	Yes
CSXJESID	N	JES identifier	Yes
CSXTDATE	N	Termination date	Yes
CSXTTIME	N	Termination time	Yes
CSXACSA	N	Amount of CSA	Yes
CXSAECS	N	Amount of ECSA	Yes
CSXASQA	N	Amount of SQA	Yes
CSXAESQ	N	Amount of ESQA	Yes

STORF - Tabular report data table ERBSTFT3

RMF builds ERBSTFT3 when using STORF as a report type.

Name	T	Description of the Variable	Report
STFDTLLN	K	Logical line number	-
STFDTPSN	K	Sequence number	-
STFPJOB	N	Jobname	Yes
STFPCLA	N	Class (A, B, O, S, or T)	Yes
STFPDMN	N	Domain number	Yes
STFPPGN	N	Performance group number	Yes
STFPSVCL	N	Service class name	Yes
STFPTOTL	N	Frame occupancy TOTAL	Yes
STFPACTV	N	Frame occupancy ACTV	Yes
STFPIDLE	N	Frame occupancy IDLE	Yes
STFPWSET	N	Active frames WSET	Yes
STFPFIXD	N	Active frames FIXED	Yes
STFPDIV	N	Active frames DIV	Yes
STFPAUXS	N	Auxiliary storage slots	Yes
STFPPGIN	N	Page-in Rate	Yes
STFPEXIN	N	Page-in rate from expanded storage	Yes
STFPSPPI	N	Shared pages page-in rate from auxiliary storage	Util
STFPTOTS	N	Total number of shared page views	Util
STFPSVIN	N	Total number of valid shared pages	Util
STFPSPVL	N	Shared pages validation rate	Util

STORR - Tabular report data table ERBSRRT3

RMF builds ERBSRRT3 when using STORR as a report type.

Name	T	Description of the Variable	Report
SRRDTLLN	K	Logical line number	-
SRRDTPSN	K	Sequence number	-
SRRVOLVC	N	Volume serial number	Yes
SRRDEVTY	N	Device type	Yes
SRRCUTY	N	Control unit type	Yes
SRREXPCT	N	Number of exposures	Yes
SRRUSVC	N	Percentage of using	Util
SRRA1VC	N	Percentage of active	Yes
SRRA2VC	N	Percentage of connect	Yes
SRRA3VC	N	Percentage of disconnect	Yes
SRRA4VC	N	Percentage of pending	Yes
SRRA5VC	N	Percentage of DLY-DB	Util
SRRA6VC	N	Percentage of DLY-CUB	Util
SRRA7VC	N	Percentage of DLY-SPB	Util
SRRSPTVC	N	Space type	Yes
SRRAUTOT	N	Average active users: TOTAL	Yes
SRRAULOC	N	Average active users: LOCAL	Yes
SRRAUSWP	N	Average active users: SWAP	Yes
SRRAUCOM	N	Average active users: COMM	Yes
SRRPDLYR	N	Delay type header	Yes
SRRPDLYP	N	Delay reason percentage	Util

STORS - Tabular report data table ERBSRST3

RMF builds ERBSRST3 when using STORS as a report type.

Name	T	Description of the Variable	Report
SRSDTLLN	K	Logical line number	-
SRSDTPSN	K	Sequence number	-
SRSPDMPG	N	Group name for graphic report • Compatibility mode: DMNxxx, PGxxx • Goal mode: WLM group name	Yes
SRSPDMN	N	Domain number	Yes
SRSPPGN	N	Performance group number	Yes
SRSPGNAM	N	Goal mode only: Name of WLM group	Yes
SRSPGTYP	N	Goal mode only: Type of WLM group	Yes
SRSPTOTU	N	Total number of users	Yes
SRSPACTU	N	Number of active users	Yes
SRS1SDEL	N	Average number delayed for ANY	Yes
SRS2SDEL	N	Average number delayed for COMM	Yes

Name	T	Description of the Variable	Report
SRS3SDEL	N	Average number delayed for LOCL	Yes
SRS4SDEL	N	Average number delayed for VIO	Util
SRS5SDEL	N	Average number delayed for SWAP	Yes
SRS6SDEL	N	Average number delayed for OUTR	Yes
SRS7SDEL	N	Average number delayed for cross memory	Util
SRS8SDEL	N	Average number delayed for hiperspace	Util
SRS9SDEL	N	Average number delayed for other reasons, including VIO, XMEM and HIPR	Yes
SRSPACTV	N	Average ACTV frames	Yes
SRSPFIXD	N	Average FIXED frames	Yes
SRSPIDLE	N	Average IDLE frames	Yes
SRSPPGIN	N	Page-in rate	Yes

SYSENQ - Tabular report data table ERBEQST3

RMF builds ERBEQST3 when using SYSENQ as a report type.

Name	T	Description of the Variable	Report
EQSDTLLN	K	Logical line number	-
EQSDTPSN	K	Sequence number	-
EQSPMAJN	N	Resource major name or resource minor name	Yes
EQSPWDEP	N	Percentage of delay for the delayed job	Yes
EQSPWJOB	N	Jobname of delayed job	Yes
EQSPWSNM	N	MVS system name of delayed job	Yes
EQSPWSTT	N	Status of delayed job	Yes
EQSPHUSP	N	Holding percentage for the holding job	Yes
EQSPHJOB	N	Jobname of holding job	Yes
EQSPHSNM	N	MVS system name of holding job	Yes
EQSPHSTT	N	Status of holding job	Yes

SYSINFO - Tabular report data table ERBSYST3

RMF builds ERBSYST3 when using SYSINFO as a report type.

Name	T	Description of the Variable	Report
SYSDTLLN	K	Logical line number	-
SYSDTPSN	K	Sequence number	-
SYSNAMVC	N	Group name • Compatibility mode: DMNxxx or PGxxx • Goal mode: WLM group name	Yes
SYSTYPVC	N	Type of WLM group (goal mode)	Yes
SYSWFLVC	N	Workflow percentage	Yes
SYSTUSVC	N	Average number of total users	Yes
SYSAUSVC	N	Average number of active users	Yes

SYSINFO data

Name	T	Description of the Variable	Report
SYSTRSVC	N	Transactions / sec	Yes
SYSAFCVC	N	Active frames percentage	Util
SYSVECVC	N	Vector utilization	Yes
SYSAUPVC	N	Average number using PROC	Yes
SYSAUDVC	N	Average number using DEV	Yes
SYSADPVC	N	Average number delayed for PROC	Yes
SYSADDVC	N	Average number delayed for DEV	Yes
SYSADSVC	N	Average number delayed for STOR	Yes
SYSADUVC	N	Average number delayed for SUBS	Yes
SYSADOVC	N	Average number delayed for OPER	Yes
SYSADEVC	N	Average number delayed for ENQ	Yes
SYSADJVC	N	Average number delayed for JES	Util
SYSADHVC	N	Average number delayed for HSM	Util
SYSADXVC	N	Average number delayed for XCF	Util
SYSADNVC	N	Average number delayed for Mount	Util
SYSADMVC	N	Average number delayed for Message	Util
SYSCPUVC	N	Percentage of CPU time (TCB+SRB) used	Util
SYSSRBVC	N	Percentage of SRB time used	Util
SYSTCBVC	N	Percentage of TCB time used	Util
SYSRSPVC	N	Average response time/transaction	Yes
SYSVELVC	N	Execution velocity	Util
SYSUGMVC	N	% using	Util
SYSUGPVC	N	% using processor	Util
SYSUGDVC	N	% using device	Util
SYSWGDVC	N	% device workflow	Util
SYSWGPVC	N	% processor workflow	Util
SYSDGMVC	N	% delay	Util
SYSUJMVC	N	Avg number users using	Util
SYSDJMVC	N	Avg number users delayed	Util
SYSDGEVC	N	% delay for ENQ	Util
SYSDGHVC	N	% delay for HSM	Util
SYSDGDVC	N	% delay for DEV	Util
SYSDGJVC	N	% delay for JES	Util
SYSDGOVC	N	% delay for OPER	Util
SYSDGPVC	N	% delay for PROC	Util
SYSDGSVC	N	% delay for STOR	Util
SYSDGUVC	N	% delay for SUBS	Util
SYSDGXVC	N	% delay for XCF	Util

SYSRTD - Tabular report data table ERBRTDT3

RMF builds ERBRTDT3 when using SYSRTD as a report type.

Name	T	Description of the Variable	Report
RTDDTLLN	K	Logical line number	-
RTDDTPSN	K	Sequence number	-
RTDSYS	N	System identifier	Yes
RTDDAT	N	Data availability indication	Yes
RTDRTQ	N	Queued time / trx	Yes
RDTRTA	N	Active time / trx	Yes
RTDRTT	N	Total response time / trx	Yes
RTDTRAN	N	Ended transactions / second	Yes
RDTSSA	N	Transaction active percentage	Yes
RTDSSR	N	Transaction ready percentage	Yes
RTDSSD	N	Transaction delay percentage	Yes
RTDEXV	N	Execution velocity percentage	Yes
RTDEXD	N	Overall delay percentage	Yes

SYSSUM - Tabular report data table ERBSUMT3

RMF builds ERBSUMT3 when using SYSSUM as a report type.

Name	T	Description of the Variable	Report
SUMDTLLN	K	Logical line number	-
SUMDTPSN	K	Sequence number	-
SUMGRP	N	Group name	Yes
SUMTYP	N	Type of WLM group	Yes
SUMIMP	N	Importance of service class period	Yes
SUMVEG	N	Execution velocity goal	Yes
SUMEVA	N	Execution velocity actual	Yes
SUMRTGT	N	Response time goal	Yes
SUMRTGP	N	Response time goal percentile	Yes
SUMRTAT	N	Response time actual	Yes
SUMRTAP	N	Response time actual percentile	Yes
SUMPFID	N	Performance index	Yes
SUMTRAN	N	Ended transactions / second	Yes
SUMARTQ	N	Queued time	Yes
SUMARTA	N	Active time	Yes
SUMARTT	N	Total response time	Yes
SUMGOA	N	Goal type	Util
SUMDUR	N	Duration	Util
SUMRES	N	Resource group name	Util
SUMSMI	N	Service rate (capacity), min.	Util
SUMSMA	N	Service rate (capacity), max.	Util

Name	T	Description of the Variable	Report
SUMSRA	N	Service rate (capacity), actual	Util

SYSWKM - Tabular report data table ERBWKMT3

RMF builds ERBWKMT3 when using SYSWKM as a report type.

Name	T	Description of the Variable	Report
WKMDTLLN	K	Logical line number	-
WKMDTPSN	K	Sequence number	-
WKMJOB	N	Jobname	Yes
WKMASI	N	Address space identification	Yes
WKMSYS	N	System identifier	Yes
WKMSRV	N	Service class name	Yes
WKMSER	N	Service percentage	Yes
WKMPRC	N	Processor using percentage	Yes
WKMVEL	N	Execution velocity percentage	Yes
WKMCAP	N	Capped delay percentage	Yes
WKMQUI	N	Address space quiesced percentage	Yes

WFEX - Tabular report data table ERBWFXT3

RMF builds ERBWFXT3 when using WFEX as a report type.

Name	T	Description of the Variable	Report
WFXDTLLN	K	Logical line number	-
WFXDTPSN	K	Sequence number	-
WFXATTR	N	Attribute	Util
WFXNAME	N	Name	Yes
WFXREASN	N	Reason	Yes
WFXDELAY	N	Delay	Yes
WFXPCAUS	N	Possible cause	Yes

XCF - Tabular report data table ERBXCFT3

RMF builds ERBXCFT3 when using XCF as a report type.

Name	T	Description of the Variable	Report
XCFDTLLN	K	Logical line number	-
XCFDTPSN	K	Sequence number	-
XCFPJOB	N	Jobname	Yes
XCFPCLA	N	Class (A, B, O, S, or T)	Yes
XCFPDMN	N	Domain	Yes
XCFPPGN	N	Performance Group	Yes
XCFPSVCL	N	Service class name	Yes
XCFPODEL	N	Overall delay	Yes

Name	T	Description of the Variable	Report
XCF1SDEL	N	Delay percentage (Path 1)	Yes
XCF1PATH	N	Path 1	Yes
XCF2SDEL	N	Delay percentage (Path 2)	Yes
XCF2PATH	N	Path 2	Yes
XCF3SDEL	N	Delay percentage (Path 3)	Yes
XCF3PATH	N	Path 3	Yes
XCF4SDEL	N	Delay percentage (Path 4)	Yes
XCF4PATH	N	Path 4	Yes

End of Programming Interface information

Graphic report parameter table ERBPTGS3

Programming Interface information

The graphic report parameter table defines the layout of graphic reports for panel display and hardcopy printing. The first part describes general information about the graphic report. The second part describes information about labels per bar. The third part describes the column layout.

The format for general information is:

Name	T	Description of the Variable	Example
PTGREPNA	K	Report name (must be specified)	DEV HSM JES
PTGRHELP	N	Name for help panel – See name convention for HELP panels	
PTGRMINY	N	Length of Y-scale, if there is no bar exceeding this length. **1** for average number of user's time, **100** for percentage values	1 100
PTGRAXTI	N	Title of the axis • Percentage of Each User's Time • Percentage of The User's Time • Average Number of Active Users	1 100
PTGRSERU	N	Selection rule for bars: **0 :** One bar corresponds to one line **1 :** One bar corresponds to one line with the sequence number 1 **2 :** One bar corresponds to the summary of logical lines **3 :** Two bar-types result from all logical lines of a logical block • Bar type 1 corresponds to sequence number 1 • Bars of bar type 2 correspond to each line of the logical block	0 1 2 3 DELAY DEV, HSM, JES DEVR, ENQR STORR
PTGRBRNM	N	Number of bar types '1' and '2', represented by the character before the last character in the following variables.	1 2

The format for labels per bar is:

Variable Name	T	Variable Description	Example
PTGRLB10	N	Number of labels per bar for bar type 1	1 2
PTGRCL1 PTGRCL2	N N	ISPF COLUMN data-table variables containing the labels for bar-type 1.	
PTGRAP1 PTGRAP2	N N	Alpha part of the labels, which will be composed by this part and the last 4 digits of the data value.	‘DMN’, ‘PG’ in DELAY
PTGRLB20	N	Number of labels per bar for bar type 2	1 2
PTGRCL3 PTGRCL4 PTGRAP3 PTGRAP4	N N N N	(corresponding to PTGRCL1) (corresponding to PTGRCL2) (corresponding to PTGRAP1) (corresponding to PTGRAP2)	

The format for columns is:

Variable Name	T	Variable Description	Example
PTGRCPNM PTGRTV1 PTGRLD1 PTGRAL1 PTGRDC1	N N N N N	Number of data columns to be selected for the bar types. = number of color-pattern-text combin. (0, 1, 2, ... 9, represented by the last character of the variable. ISPF Column Table variable. This variable contains a specific data value of the tabular report after a TBGET to a row of the Data Column Table. (Must be specified) Legend ID, to specify a particular color-pattern-text combination of the Color-Pattern Table. The ID specifies the legend (color, pattern and subheader) for this data value. Transformation ID 0 : don’t divide 1 : divide by 10 2 : divide by 100 bartype col ; If ‘0’, the data value 0 : reports value in both bar types 1 : reports value in first bar type 2 : reports value in second bar type	0 1 ... 9 1 2 ... see color-pattern option table 0 2 0 1 2
PTGRTV2 PTGRLD2 PTGRAL2 PTGRDC2	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV3 PTGRLD3 PTGRAL3 PTGRDC3	N N N N	(corresponding to ptgrtv1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	

Graphic layout

Variable Name	T	Variable Description	Example
PTGRTV4 PTGRLD4 PTGRAL4 PTGRDC4	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV5 PTGRLD5 PTGRAL5 PTGRDC5	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV6 PTGRLD6 PTGRAL6 PTGRDC6	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV7 PTGRLD7 PTGRAL7 PTGRDC7	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV8 PTGRLD8 PTGRAL8 PTGRDC8	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV9 PTGRLD9 PTGRAL9 PTGRDC9	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV10 PTGRLD10 PTGRAL10 PTGRDC10	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	

End of Programming Interface information

RMF Phase driver table ERBPHDS3

Programming Interface information

The phase driver table has rows for each command and selection.

Variable Name	T	Variable Description
PHDREPNA	K	Name of the command or the long form of the report selection.
PHDREPSE	N	Selection string to be created. This string will be passed to the primary option panel to perform the command function.
PHDRPH1	N	Function to be performed for Phase 1. The string if not null, will be selected.
PHDRPH2	N	Function to be performed for Phase 2. The string if not null, will be selected.
PHDRPH3	N	Function to be performed for Phase 3. The string if not null, will be selected.
PHDRPH4	N	Function to be performed for Phase 4. The string if not null, will be selected.
PHDRET1	N	Return code passed from Phase 1. The Phase 2 and Phase 3 are executed only if the return code from this Phase is zero.
PHDRET2	N	Return code passed from Phase 2. The Phase 3 is executed only if the return code from this Phase is zero.
PHDRET3	N	Return code passed from Phase 3.
PHDRET4	N	Return code passed from Phase 5.
PHDRTAB1	N	Name of the ISPF table created by Phase 1. This table is input for Phase 2.
PHDRTAB2	N	Name of the ISPF table created by Phase 2. This table is input to Phase 3.

This table lists the report commands, selections, and the variables used for each phase (1,2,3,4). Phase 2 and 4 are null.

PHDREPNA	PHDREPSE	PHDRPH1	PHDRPH3	PHDRTAB1
CACHDET	S.9	PGM(ERB3RPH1) PARM(CACHDET)	PGM(ERB3RDSP)	ERBCADT3
CACHSUM	S.8	PGM(ERB3RPH1) PARM(CACHSUM)	PGM(ERB3RDSP)	ERBCAST3
CFACT	S.7	PGM(ERB3RPH1) PARM(CFACT)	PGM(ERB3RDSP)	ERBCFAT3
CFOVER	S.5	PGM(ERB3RPH1) PARM(CFOVER)	PGM(ERB3RDSP)	ERBCFOT3
CFSYS	S.6	PGM(ERB3RPH1) PARM(CFSYS)	PGM(ERB3RDSP)	ERBCFST3
CHANNEL	3.12	PGM(ERB3RPH1) PARM(CHANNEL)	PGM(ERB3RDSP)	ERBCHAT3
CPC	1.3	PGM(ERB3RPH1) PARM(CPC)	PGM(ERB3RDSP)	ERBCPCT3
DELAY	1.4	PGM(ERB3RPH1) PARM(DELAY)	PGM(ERB3RDSP)	ERBJDET3
DEV	3.2	PGM(ERB3RPH1) PARM(DEV)	PGM(ERB3RDSP)	ERBDEVT3
DEVR	3.3	PGM(ERB3RPH1) PARM(DEVR)	PGM(ERB3RDSP)	ERBDVRT3
DSINDEX	S.D	PGM(ERB3RHP1) PARM(DSINDEX)	PGM(ERB3RDSP)	ERBDSIT3
DSND	3.3A	PGM(ERB3RPH1) PARM(DSND)	PGM(ERB3RDSP)	ERBDNDT3
DSNJ	2.1A	PGM(ERB3RPH1) PARM(DSNJ)	PGM(ERB3RDSP)	ERBDNJT3
DSNV	3.3B	PGM(ERB3RPH1) PARM(DSNV)	PGM(ERB3RDSP)	ERBDNVT3

Phase driver

PHDREPNA	PHDREPSE	PHDRPH1	PHDRPH3	PHDRTAB1
ENCLAVE	1.6	PGM(ERB3RPH1) PARM(ENCLAVE)	PGM(ERB3RDSP)	ERBENCT3
ENQ	3.4	PGM(ERB3RPH1) PARM(ENQ)	PGM(ERB3RDSP)	ERBENQT3
ENQR	3.5	PGM(ERB3RPH1) PARM(ENQR)	PGM(ERB3RDSP)	ERBEQRT3
GROUP	1.5	PGM(ERB3RPH1) PARM(GROUP)	PGM(ERB3RDSP)	ERBGRTT3
HSM	4.1	PGM(ERB3RPH1) PARM(HSM)	PGM(ERB3RDSP)	ERBHSMT3
IOQ	3.13	PGM(ERB3RPH1) PARM(IOQ)	PGM(ERB3RDSP)	ERBIOQT3
JES	4.2	PGM(ERB3RPH1) PARM(JES)	PGM(ERB3RDSP)	ERBJEST3
JOB	2.5	PGM(ERB3RPH1) PARM(JOB)	PGM(ERB3RDSP)	ERBJDJT3
OPD	1.7	PGM(ERB3RPH1) PARM(OPD)	PGM(ERB3RDSP)	ERBOPDT3
PROC	3.1	PGM(ERB3RPH1) PARM(PROC)	PGM(ERB3RDSP)	ERBPRCT3
RLSDS	S.11	PGM(ERB3RPH1) PARM(RLSDS)	PGM(ERB3RDSP)	ERBVRDT3
RLSLRU	S.12	PGM(ERB3RPH1) PARM(RLSLRU)	PGM(ERB3RDSP)	ERBVRLT3
RLSSC	S.10	PGM(ERB3RPH1) PARM(RLSSC)	PGM(ERB3RDSP)	ERBVRST3
STOR	3.6	PGM(ERB3RPH1) PARM(STOR)	PGM(ERB3RDSP)	ERBSTRT3
STORC	3.10	PGM(ERB3RPH1) PARM(STORC)	PGM(ERB3RDSP)	ERBCSUT3
STORCR	3.11	PGM(ERB3RPH1) PARM(STORCR)	PGM(ERB3RDSP)	ERBCRST3
STORF	3.7	PGM(ERB3RPH1) PARM(STORF)	PGM(ERB3RDSP)	ERBSTFT3
STORR	3.8	PGM(ERB3RPH1) PARM(STORR)	PGM(ERB3RDSP)	ERBSRRT3
STORS	3.9	PGM(ERB3RPH1) PARM(STORS)	PGM(ERB3RDSP)	ERBSRST3
SYSENQ	S.4	PGM(ERB3RPH1) PARM(SYSENQ)	PGM(ERB3RDSP)	ERBEQST3
SYSINFO	1.2	PGM(ERB3RPH1) PARM(SYSINFO)	PGM(ERB3RDSP)	ERBSYST3
SYSRTD	S.2	PGM(ERB3RPH1) PARM(SYSRTD)	PGM(ERB3RDSP)	ERBRTDT3
SYSSUM	S.1	PGM(ERB3RPH1) PARM(SYSSUM)	PGM(ERB3RDSP)	ERBSUMT3
SYSWKM	S.3	PGM(ERB3RPH1) PARM(SYSWKM)	PGM(ERB3RDSP)	ERBWKMT3
WFEX	1.1	PGM(ERB3RPH1) PARM(WFEX)	PGM(ERB3RDSP)	ERBWFXT3
XCF	4.3	PGM(ERB3RPH1) PARM(XCF)	PGM(ERB3RDSP)	ERBXCFT3

End of Programming Interface information

Chapter 8. Diagnosis reference

Diagnosing problems in RMF

Before using this book, you should:

- Be familiar with RMF. See *z/OS RMF User's Guide* for specific information about RMF sessions.
- Understand the information in the *z/OS MVS Diagnosis: Procedures*.

Identifying problems

The following table lists each problem type and its meaning. The table directs you to a diagnostic procedure for that problem type or to another book.

Table 8-1. Problem Types

For these problem types:	See the following:
Abend X'0D5': RMF detected an address space identifier (ASID) that was not valid.	"Diagnosing abend 0D5" on page 8-2
Abend X'0FE': An error occurred, ending the Monitor I ZZ session.	"Diagnosing abend 0FE" on page 8-3
Abend unexpected by RMF: A module with the prefix ERB abnormally ended with a system completion code other than the codes listed above.	"Diagnosing an abend unexpected by RMF" on page 8-4
Message with ERB prefix.	"Diagnosing a message with an ERB prefix" on page 8-5
Message with CEE or EDC prefix.	"Diagnosing a message with a CEE/EDC prefix" on page 8-6
Abend with a user completion code issued by a module with an ERB prefix.	*z/OS RMF Messages and Codes*
Incorrect output: Unusually large or small numbers appeared in an RMF report.	"Diagnosing incorrect output" on page 8-7
Documentation error: An error in RMF documentation was detected.	"Diagnosing a documentation error" on page 8-8
Empty Monitor III JES delays report: A Monitor III JES delays report was displayed that contained no data. The headings appeared, but the fields were blank.	"Diagnosing an empty Monitor III JES Delays report" on page 8-9

Diagnosing abend 0D5

Use this procedure when a module with the prefix ERB abnormally ends with the system completion code X'0D5'. RMF detected an ASID that was not valid.

Table 8-2. Diagnostic Procedure for Abend 0D5

Diagnostic Procedure	References
1. Obtain the SYS1.LOGREC error record and format it with EREP to obtain a detail edit report.	*EREP User's Guide* for EREP formatting *z/OS MVS Diagnosis: Tools and Service Aids* to read the SYS1.LOGREC record *z/OS MVS Diagnosis: Tools and Service Aids*
2. Check the SYS1.LOGREC error record to determine if the abend occurred in module ERBMFPVS. If the abend did not occur in module ERBMFPVS, continue with step 4. Otherwise proceed with the next step.	*z/OS MVS Diagnosis: Tools and Service Aids* to read the SYS1.LOGREC record *z/OS MVS Diagnosis: Tools and Service Aids*
3. Obtain the SVC dump for the abend.	*z/OS MVS Diagnosis: Procedures*
4. Format the dump with the IPCS VERBEXIT LOGDATA subcommand to see the system diagnostic work area (SDWA) and the search argument. In the dump, find the offset of the failing instruction into the module. If analyzing the dump at a terminal, use the IPCS WHERE subcommand to locate the failing instruction and to find its offset into the module.	*z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA and WHERE subcommands
5. Determine if the ASID is not valid. Either: • The address space does not exist • The address space is swapped out. Correct the problem and reinitialize RMF. If you cannot determine why the ASID is not valid, continue with the next step.	
6. Develop a search argument consisting of: • Programmer identifier: IDS/566527404 • System abend code: AB/S00D5 • CSECT name: RIDS/cccccccc • Load module name: RIDS/cccccccc#L Use the search argument to search problem reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.	Developing a search argument for RMF *z/OS MVS Diagnosis: Procedures*
7. Report the problem to IBM. Provide the following problem data: • Load module name and level • CSECT name and level • Offset of the failing instruction into the module • SVC dump (softcopy) • Search argument • Current RMF Monitor I options • RMF version and release • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing abend 0FE

Use this procedure when RMF ends with a system completion code X'0FE'. RMF detected an error while sampling data about the state of the system.

Table 8-3. Diagnostic Procedure for Abend 0FE

Diagnostic Procedure	References
1. Obtain the SYS1.LOGREC error record and format it with EREP to obtain a detail edit report.	*EREP User's Guide* for EREP formatting *z/OS MVS Diagnosis: Tools and Service Aids* to read the SYS1.LOGREC record *z/OS MVS Diagnosis: Tools and Service Aids*
2. Locate the 0FE entry in the SYS1.LOGREC error record. The entry preceding X'0FE' indicates the module that abended.	*z/OS MVS Diagnosis: Tools and Service Aids*
3. Set a SLIP trap for the abend code preceding the X'0FE' entry in the SYS1.LOGREC error record. Request the SLIP to produce an SVC dump. If you cannot reproduce the situation, continue with step 5.	*z/OS MVS Diagnosis: Procedures*
4. Format the dump produced by the SLIP trap with the IPCS VERBEXIT LOGDATA subcommand and keep as softcopy. In the dump, find the offset of the failing instruction into the module. If analyzing the dump at a terminal, use the IPCS WHERE subcommand to locate the failing instruction and to find its offset into the module.	*z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA and WHERE subcommands
5. Develop a search argument consisting of: • Program identifier: PIDS/566527404 • Load module name: RIDS/cccccccc#L • CSECT name: RIDS/cccccccc • System abend code: AB/S00FE • SLIP trap abend code: AB/S0xxx Use the search argument to search problem reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.	Developing a search argument for RMF *z/OS MVS Diagnosis: Procedures*
6. Report the problem to IBM. Provide the following problem data: • Load module name and level • CSECT name and level • Offset of the failing instruction into the module • Registers at time of abend • SVC dump produced by the SLIP trap (softcopy) • SYS1.LOGREC (softcopy) • Search argument • Current RMF Monitor I options • RMF version and release • Console log of this situation including related ERBxxxI messages • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing an abend unexpected by RMF

Use this procedure when a module with the prefix ERB abnormally ends with a system completion code other than the codes listed in the table under "Identifying problems" on page 8-1.

Table 8-4. Diagnostic Procedure for an Abend Unexpected by RMF

Diagnostic Procedure	References
1. Look at the explanation for the abend code and any accompanying reason code. Take the recommended actions.	*z/OS MVS System Codes* for an explanation of the abend code
2. Obtain messages accompanying the abend. Look at their explanations and take any recommended actions.	Please refer to Using LookAt to look up message explanations for getting the fastest message explanation.
3. Obtain the SYS1.LOGREC error record, the dump, or both for the abend (softcopy).	*z/OS MVS Diagnosis: Tools and Service Aids*
4. Format the dump or SYS1.LOGREC record to see the SDWA and the search argument: • Format an SVC dump or SYSMDUMP ABEND dump with the IPCS VERBEXIT LOGDATA subcommand. • Format a SYS1.LOGREC record with EREP to obtain a detail edit report. Find the heading VARIABLE RECORDING AREA (SDWAVRA). Note the SDWAVRA keys, lengths, and contents.	*z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA subcommand *EREP User's Guide* for EREP formatting *z/OS MVS Diagnosis: Tools and Service Aids* *z/OS MVS Diagnosis: Tools and Service Aids*
5. In the dump, find the offset of the failing instruction within the CSECT of the load module. If analyzing the dump at a terminal, use the IPCS WHERE subcommand to locate the failing instruction and to find its offset into the module.	*z/OS MVS IPCS Commands* for the WHERE subcommand
6. Use the WRITELOG system command to print the system log (keep as softcopy). The system log shows all system messages and commands issued. Make sure you print the log for the time period when the abend occurred.	*z/OS MVS System Commands* for the WRITELOG command
7. Develop a search argument consisting of: • Program identifier: PIDS/566527404 • System abend code: AB/S0xx • Abend reason code: PRCS/xxxxxxxx • Message identifier: MS/cccnnns • CSECT: RIDS/cccccccc • Load module name: RIDS/cccccccc#L Use the search argument to search problem reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.	Developing a search argument for RMF *z/OS MVS Diagnosis: Procedures*
8. Report the problem to IBM. Provide the following problem data: • Accompanying messages • The dump, SYS1.LOGREC error record, or both (softcopy) • SDWAVRA keys, lengths, and contents • Hardcopy (or better softcopy) of console log • Search argument • Offset of the failing instruction into the module • Current RMF options for session running • RMF version and release • Load module name and level • CSECT name and level • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing a message with an ERB prefix

Use this procedure when you receive a message with an ERB prefix.

Table 8-5. Diagnostic Procedure for a Message with an ERB Prefix

Diagnostic Procedure	References
1. Determine which session was running when the message was issued. • If the message was issued within an RMF ISPF reporter session, use the help facility RMF supplies for an explanation of the message and user response. • If the message was received on the main operator console, it is a system message. If you need assistance with a message, continue with the next step.	Press the HELP key (PF1) for an explanation of the message. These message explanations are also listed in the *z/OS RMF Messages and Codes* book. For an explanation and an appropriate operator response for system messages, please refer to Using LookAt to look up message explanations.
2. Report the problem to IBM. Provide the following problem data: • Message number • RMF version and release • RMF session: Monitor I, II, III or Postprocessor • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing a message with a CEE/EDC prefix

Use this procedure when you receive a message with a CEE or EDC prefix.

These message are created by integrated LE/370 routines during a Postprocessor session.

Table 8-6. Diagnostic Procedure for a Message with a CEE/EDC Prefix

Diagnostic Procedure	**References**
1. Increase the region size for the Postprocessor job. If the problem is not solved, continue with the next step.	See the *z/OS RMF User's Guide* for details. See *z/OS Language Environment Debugging Guide* for an explanation of the message.
2. Report the problem to IBM. Provide the following problem data: • Message number • RMF version and release • RMF session: Postprocessor • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing incorrect output

Use this procedure when unusually large or small numbers appear in any RMF reports. Inaccurate input might have been given to RMF from another system component.

Table 8-7. Diagnostic Procedure for Incorrect Output

Diagnostic Procedure	References
1. Obtain a hardcopy of the report that has unusual numbers. If the problem is a Monitor III JES delays report which has no data, see "Diagnosing an empty Monitor III JES Delays report" on page 8-9.	*z/OS RMF User's Guide*
2. If you are running a Monitor I or Monitor II session or the Postprocessor, and you are collecting SMF data, print the contents of the SMF records which generated *this* report (and keep them as softcopy). If you are running a Monitor III session, keep any VSAM data sets used to hold data during the session. If the problem is with a report produced by a Monitor III SMF record, print the contents of that SMF record (and keep it as softcopy).	*z/OS RMF User's Guide*
3. Report the problem to IBM. Provide the following problem data: • Hardcopy (or better softcopy) of the report • Accompanying messages • RMF version and release • RMF session: Monitor I, II, III or Postprocessor • SMF record contents • VSAM data set contents if running a Monitor III session • Current RMF options • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing a documentation error

Use this procedure when you find an error in RMF documentation.

Table 8-8. Diagnostic Procedure for a Documentation Error

Diagnostic Procedure	References
1. If you have a problem with an RMF publication use the Reader's Comment Form of that book to report the documentation error. Be specific when reporting the error. If there is no Reader's Comment Form, send your description of the documentation error to the address listed in the edition notice of the book. Include the following information: • Publication title • Publication order number • Page number containing the problem • Thorough description of the problem. If the problem is with a form of documentation other than a publication, continue with the next step.	See the Reader's Comment Form of this book for an example.
2. If the problem was with an RMF ISPF panel, type PANELID on the command line to display the panel ID. Use the Reader's Comment Form of *z/OS RMF Report Analysis* to report the problem. Include the following information: • RMF version and release • Panel ID • Thorough description of the problem.	See the Reader's Comment Form of this book for an example.
3. Report the problem to the IBM Support Center *only* in the following situations: • The correction to the documentation is needed to prevent a severe problem. • You are not sure if the problem is a documentation error or product error. If you report the problem to the IBM Support Center, provide the following data: • RMF version and release • The name and order number of the publication you are using and the page containing the error • If the error was on an online panel, provide the panel ID. Type PANELID on the command line to display the panel ID. • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Diagnosing an empty Monitor III JES Delays report

Use this procedure when no data appears in a Monitor III JES delays report.

If there are no jobs delayed by JES, RMF generates an empty Monitor III JES delays report. In this case, an empty report is not considered a problem.

Table 8-9. Diagnostic Procedure for Empty Monitor III JES Delays Report

Diagnostic Procedure	References
1. Find out if the JES2 or JES3 control blocks have changed at your installation. If so, Monitor III modules might be affected.	For the RMF/JES interface module installation procedures, see *z/OS Program Directory*
2. If the JES control blocks have changed, you must reassemble and link-edit the RMF source modules that map offsets to the JES control block fields sampled by Monitor III using the new JES macros. If the problem occurs again, continue with the next step.	
3. Report the problem to IBM. Provide the following problem data: • Accompanying messages • RMF version and release • Installed JES level • Hardcopy of the report • VSAM data set contents • Name and level of the operating system with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	Reporting a problem to IBM

Obtaining a dump from Monitor II or Monitor III

If an error occurs in the RMF Monitor II or Monitor III reporter session, you will be prompted whether you want to write a dump. Follow the steps below to obtain a dump.

Table 8-10. Procedure for Obtaining a Dump

Diagnostic procedure	**References**
1. Enter in the command line: **TSO FREE FI(SYSUDUMP SYSABEND)** You can ignore messages, for example: *IKJ56247I FILE xxxxxxxx NOT FREED, IS NOT ALLOCATED*	
2. Enter in the command line: **TSO ALLOC FI(SYSMDUMP) DA(dsname) NEW SP(200 200) CYL REUSE REL** If the command does not fit into the command line, please, start split-screen mode to enter the command.	
3. Then answer ″Y″ in the dump request panel: *Would you like a dump? Enter Y or N. ===>***Y**	
4. The system now writes an unformatted dump to the data set just allocated. This may take some time. When it is finished, the system issues message: *IEA993I SYSMDUMP TAKEN TO dsname*	
5. The dump can now be processed with IPCS.	

Developing a search argument for RMF

You or IBM can use a search argument to search a problem reporting data base to look for a problem similar to the one you encountered. If the problem was reported previously, a problem reporting data base contains information about the problem and, possibly, a fix. See *z/OS MVS Diagnosis: Procedures* for detailed descriptions of formatting search arguments and searching data bases.

The following table shows symptoms for RMF search arguments. The table summarizes the symptoms recommended in the diagnostic procedures in Diagnosing problems in RMF.

Use the **free-format** if your installation has a free-format search tool, such as INFO/System with the INFO/MVS data base.

Use the **structured** format if your installation has a structured format search tool, such as INFO/Management or INFO/Access.

Table 8-11. Search Arguments from Diagnostic Procedures

Description	Free Format	Structured Format	Examples
System abend code	abendhhh	AB/S0hhh	abend0D5 AB/S00D5
User completion code	abenddddd	AB/Udddd	abend1207 AB/U1207
Message identifier	msgccccccc	MS/ccccccc	msgERB251I MS/ERB251I
Load module name	cccccccc	RIDS/cccccccc#L	ERB3GMFC RIDS/ERB3GMFC#L
CSECT name (object module)	cccccccc	RIDS/cccccccc	ERB3GISS RIDS/ERB3GISS
Return code	rchhhhhhhh	PRCS/hhhhhhhh	00000020 PRCS/00000020
Program identifier The first four characters identify the product MVS and the last five characters identify RMF.	ccccccccc	PIDS/ccccccccc	566527404 PIDS/566527404

Reporting a problem to IBM

The following tables identify the information you need to collect before calling IBM to report an RMF problem. When you report a problem, you need to describe your system and the problem you experienced. The IBM Support Center personnel uses this information to see if the problem is already known to IBM, check whether a fix is available, or determine how to correct the problem.

Table 8-12 lists the problem data you need to collect before calling IBM to report a problem with an abend X'0D5'.

Table 8-12. Checklist for Reporting a Problem with an Abend 0D5

Problem data	**Example or reference**	**Information collected**
Load module name and level[1]	ERB*xxxxx* U*nnnnnn*[2]	
CSECT name and level	ERB*xxxxx* U*nnnnnn*	
Offset of the failing instruction into the module	X'090AF'	
SVC dump (SYSMDUMP as softcopy) *z/OS MVS Diagnosis: Procedures*		
Search argument	"Developing a search argument for RMF" on page 8-11 *z/OS MVS Diagnosis: Procedures*	
Current RMF options	Use the RMF DISPLAY command to display the current options on the operator console. *z/OS RMF User's Guide*	
RMF version and release	z/OS V1R2 RMF	
Program temporary fix (PTF) numbers	U*nnnnnn*	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		
Note: 1. The level can be U*nnnnnn*, HRM*nnnn*, or JRM*nnnn*. 2. *xxxxx* and *nnnnnn* are just placeholders. The actual data can be obtained from the dump.		

Table 8-13 lists the problem data you need to collect before calling IBM to report a problem with an abend X'0FE'.

Table 8-13. Checklist for Reporting a Problem with an Abend 0FE

Problem data	**Example or reference**	**Information collected**
Load module name and level[1]	ERB*xxxxx* U*nnnnnn*[2]	
CSECT name and level	ERB*xxxxx* U*nnnnnn*	
Offset of the failing instruction into the module	X'090AF'	
SVC dump produced by the SLIP trap (softcopy)	*z/OS MVS Diagnosis: Procedures*	
SYS1.LOGREC (softcopy)	*z/OS MVS Diagnosis: Tools and Service Aids*	
Console log (softcopy)		
Search argument	"Developing a search argument for RMF" on page 8-11 *z/OS MVS Diagnosis: Procedures*	
Current Monitor I options	Use the RMF DISPLAY command to display the current options on the operator console. *z/OS RMF User's Guide*	
RMF version and release	z/OS V1R2 RMF	
Program temporary fix (PTF) numbers	U*nnnnnn*	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		

Note:
1. The level can be U*nnnnnn*, HRM*nnnn*, or JRM*nnnn*.
2. *xxxxx* and *nnnnnn* are just placeholders. The actual data can be obtained from the dump.

Reporting a problem

Table 8-14 lists the problem data you need to collect before calling IBM to report a problem with an unexpected abend.

Table 8-14. Checklist for Reporting a Problem with an Unexpected Abend

Problem data	**Example or reference**	**Information collected**
Load module name and level[1]	ERB*xxxxx* U*nnnnnn*[2]	
CSECT name and level	ERB*xxxxx* U*nnnnnn*	
Messages accompanying the problem, including the message identifier and variable data in the message text	Identifier: ERB259I Text: EXCEPTION REPORTING TERMINATED *z/OS RMF Messages and Codes* Please refer to Using LookAt to look up message explanations for getting the fastest message explanation.	
SVC dump (SYSMDUMP as softcopy)	*z/OS MVS Diagnosis: Procedures*	
SYS1.LOGREC error record (softcopy)	*z/OS MVS Diagnosis: Procedures*	
SDWAVRA keys, lengths, and contents	*z/OS MVS Diagnosis: Procedures*	
System log to show system messages and commands (softcopy)	*z/OS MVS System Commands* for WRITELOG command	
Search argument	"Developing a search argument for RMF" on page 8-11 *z/OS MVS Diagnosis: Procedures<*	
In case of a Monitor III error, provide the contents of the VSAM data set belonging to this problem		
Offset of the failing instruction into the module	X'090AF'	
Current RMF options for session running	Use the RMF DISPLAY command to display the current options on the operator console. *z/OS RMF User's Guide*	
RMF version and release	z/OS V1R2 RMF	
Program temporary fix (PTF) numbers	U*nnnnnn*	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		

Note:
1. The level can be U*nnnnnn*, HRM*nnnn*, or JRM*nnnn*.
2. *xxxxx* and *nnnnnn* are just placeholders. The actual data can be obtained from the dump.

Table 8-15 lists the problem data you need to collect before calling IBM to report a problem with an ERB or CEE/EDC message.

Table 8-15. Checklist for Reporting a Problem with an ERB CEE/EDC Message

Problem data	**Example or reference**	**Information collected**
Message number	ERB671I	
RMF version and release	z/OS V1R2 RMF	
RMF session	Monitor I, II, III or Postprocessor	
Program temporary fix (PTF) numbers	U*nnnnnn*[1]	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		
Note: 1. *nnnnnn* is just a placeholder. The actual data can be obtained from the dump.		

Reporting a problem

Table 8-16 lists the problem data you need to collect before calling IBM to report a problem with incorrect output.

Table 8-16. Checklist for Reporting a Problem with Incorrect Output

Problem data	**Example or reference**	**Information collected**
Hardcopy of report	*z/OS RMF User's Guide*	
Accompanying messages	*z/OS RMF Messages and Codes* Please refer to Using LookAt to look up message explanations for getting the fastest message explanation.	
RMF version and release	z/OS V1R2 RMF	
RMF session	Monitor I, II, III or Postprocessor	
SMF record contents	*z/OS RMF User's Guide*	
Unloaded VSAM data set contents if running a Monitor III session	*z/OS RMF User's Guide*	
Current RMF options	Copy from SYS1.PARMLIB	
Program temporary fix (PTF) numbers	U*nnnnnn*[1]	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		
Note: 1. *nnnnnn* is just a placeholder. The actual data can be obtained from the dump.		

Table 8-17 lists the problem data you need to collect before calling IBM to report a problem with documentation.

Table 8-17. Checklist for Reporting a Documentation Error

Problem data	**Example or reference**	**Information collected**
Publication title	*z/OS RMF Messages and Codes*	
Publication order number	SC33-7993	
Page number containing problem	3-45	
Panel ID	ERB3PRM	
RMF version and release	z/OS V1R2 RMF	
Program temporary fix (PTF) numbers	U*nnnnnn*[1]	
Thorough description of the problem.		
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		
Note: 1. *nnnnnn* is just a placeholder. The actual data can be obtained from the dump.		

Reporting a problem

Table 8-18 lists the problem data you need to collect before calling IBM to report a problem with a Monitor III JES delays report.

Table 8-18. Checklist for Reporting a Problem with a Monitor III JES Delays Report

Problem data	**Example or reference**	**Information collected**
Accompanying messages	*z/OS RMF Messages and Codes* Please refer to Using LookAt to look up message explanations for getting the fastest message explanation.	
Installed JES level	JES3 Release x.x	
Hardcopy of report	*z/OS RMF User's Guide*	
Unloaded VSAM data set contents	*z/OS RMF User's Guide*	
RMF version and release	z/OS V1R2 RMF	
Program temporary fix (PTF) numbers	U*nnnnnn*[1]	
Other problem data developed while using the *z/OS MVS Diagnosis: Procedures* book and the diagnostic procedures in Chapter 1		
Note: 1. *nnnnnn* is just a placeholder. The actual data can be obtained from the dump.		

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS™ enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Glossary

This glossary contains chiefly definitions of terms used in this book, but some more general RMF and MVS terms are also defined.

Words that are set in *italics* in the definitions are terms that are themselves defined in the glossary.

A

APPC/MVS. Advanced program-to-program communication

ASCH address space. APPC transaction scheduler address space

AS. *Address space*

address space. That part of MVS main storage that is allocated to a job.

auxiliary storage (AUX). All addressable storage, other than main storage, that can be accessed by means of an I/O channel; for example storage on direct access devices.

B

background session. In RMF, a monitor session that is started and controlled from the operator console. Contrast with *interactive session*

balanced systems. To avoid bottlenecks, the system resources (CP, I/O, storage) need to be balanced.

basic mode. A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

bottleneck. A system resource that is unable to process work at the rate it comes in, thus creating a queue.

C

callable services. Parts of a program product that have a published external interface and can be used by application programs to interact with the product.

captured storage. See shared page group.

capture ratio. The ratio of reported CPU time to total used CPU time.

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations.

central processor complex (CPC). A physical collection of hardware that consists of central storage, one or more central processors, timers, and channels.

channel path. The channel path is the physical interface that connects control units and devices to the CPU.

CICS. Customer Information Control System

compatibility mode. The implicit state of an MVS system when no workload manager service policies are in effect. Contrast with *goal mode*.

contention. Two or more incompatible requests for the same resource. For example, contention occurs if a user requests a resource and specifies exclusive use, and another user requests the same resource, but specifies shared use.

coupling facility. See *Cross-system Extended Services/Coupling Facility*.

CP. *Central processor*

criteria. Performance criteria set in the WFEX report options. You can set criteria for all report classes (PROC, SYSTEM, TSO, and so on).

CPU speed. Measurement of how much work your CPU can do in a certain amount of time.

cross-system coupling facility (XCF). A component of MVS that provides functions to support cooperation between authorized programs running within a *sysplex*.

Cross-system Extended Services/Coupling Facility (XES/CF). Provides services for MVS systems in a sysplex to share data on a coupling facility (CF).

CS. Central storage

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining data bases.

cycle. In RMF, the time at the end of which one sample is taken. Varies between 50 ms and 9999 ms. See also *sample*.

D

data sample. See *sample*

DCM. See *Dynamic Channel Path Management*

delay. The delay of an address space represents a job that needs one or more resources but that must wait because it is contending for the resource(s) with other users in the system.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data. Usually: a magnetic disk device.

DLY. Delay

DMN. Domain

domain. In compatibility mode, an optional method for setting bounds for the amount of service to be granted to a particular service class.

DP. Dispatching priority

dynamic channel path management. Dynamic channel path management provides the capability to dynamically assign channels to control units in order to respond to peaks in demand for I/O channel bandwidth. This is possible by allowing you to define pools of so-called floating channels that are not related to a specific control unit. With the help of the Workload Manager, channels can float between control units to best service the work according to their goals and their importance.

E

EMIF. ESCON multiple image facility

enclave. An enclave is a group of associated dispatchable units. More specifically, an enclave is a group of SRB routines that are to be managed and reported on as an entity.

EPDM. Enterprise Performance Data Manager/MVS

ES. Expanded storage

ESCON multiple image facility (EMIF). A facility that allows channels to be shared among PR/SM logical partitions in an ESCON environment.

execution velocity. A measure of how fast work should run when ready, without being delayed for processor or storage access.

exception reporting. In RMF, the reporting of performance measurements that do not meet user-defined criteria. Shows potential performance problems explicitly, thus avoiding the need for constant monitoring.

expanded storage (ES). (1) An extension of processor storage. (2) Optional high-speed storage that transfers 4KB pages to and from central storage.

G

generalized trace facility (GTF). A service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

GO mode. In RMF, the Monitor III mode in which the screen is updated with the interval you specified in your session options. The terminal cannot be used for anything else when it is in GO mode. See also *mode*.

goal mode. The implicit mode of an MVS system that has active service policies and performance goals defined by the workload manager. Contrast with *compatibility mode*.

graphic mode. In RMF Monitor III, the mode which presents the performance data from the system in graphic format using the GDDM product. Contrast with *tabular mode*.

GTF. generalized trace facility

H

high-speed buffer (HSB). A cache or a set of logically partitioned blocks that provides significantly faster access to instructions and data than provided by central storage.

HS. hiperspace

HSB. High-speed buffer

HSM. Hierarchical Storage Manager

I

IMS. Information Management System

Information Management System (IMS). A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS/VS.

installation performance specification (IPS). In MVS, a set of installation-supplied control information used by the system workload manager. An IPS includes performance group definitions, performance objectives, and coefficients used to establish the service rate. See also service rate.

interactive session. In RMF, a monitor display-session that is controlled from the display terminal. Contrast with *background session*.

J

JES. Job Entry Subsystem

L

LCU. Logical control unit

License Manager. The IBM License Manager is the base for a new software pricing model. It allows vendors to enable their products for licensed software management by customers and is the basic tool IBM will use to implement the Workload License Charges pricing model on z900 servers.

logically partitioned (LPAR) mode. A central processor mode that is available on the Configuration frame when using the PR/SM feature. It allows an operator to allocate processor unit hardware resources among logical partitions. Contrast with *basic mode*.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (LPAR) mode*.

LP. Logical partition

LPAR. Logically partitioned (mode)

LPAR cluster. An LPAR cluster is the subset of the systems that are running as LPARs on the same CEC. Based on business goals, WLM can direct PR/SM to enable or disable CP capacity for an LPAR, without human intervention.

M

migration rate. The rate (pages/second) of pages being moved from expanded storage through central storage to auxiliary storage.

mintime. The smallest unit of sampling in Monitor III. Specifies a time interval during which the system is sampled. The data gatherer combines all samples gathered into a set of samples. The set of samples can be summarized and reported by the reporter.

mode. Monitor III can run in various modes: GO mode (see *GO mode*) and STOP mode, which is the default mode. See also *graphic mode* and *tabular mode*.

MPL. Multiprogramming level

O

OMVS. Reference to z/OS UNIX System Services

P

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PDS. partitioned data set

performance management. (1) The activity which monitors and allocates data processing resources to applications according to goals defined in a service level agreement or other objectives. (2) The discipline that encompasses collection of performance data and tuning of resources.

performance group. Group of work with the same performance objectives managed by the SRM.

PG. Performance group

PGN. Performance group number

PR/SM. Processor Resource/Systems Manager

Processor Resource/Systems Manager (PR/SM). The feature that allows the processor to run several operating systems environments simultaneously and provides logical partitioning capability. See also *LPAR*.

R

range. The time interval you choose for your report.

Resident time. The time the address space was swapped in, in units of seconds.

S

sample. Once in every cycle, the number of jobs waiting for a resource, and what job is using the resource at that moment, are gathered for all resources of a system by Monitor III. These numbers constitute one sample.

SCP. System control program

seek. The DASD arm movement to a cylinder. A seek can range from the minimum to the maximum seek time of a device. In addition, some I/O operations involve multiple imbedded seeks where the total seek time can be more than the maximum device seek time.

service class. In Workload Manager, a subdivision of a *workload*. Performance goals and capacity boundaries are assigned to service classes.

service level agreement (SLA). A written agreement of the information systems (I/S) service to be provided to the users of a computing installation.

Service Level Reporter (SLR). An IBM licensed program that provides the user with a coordinated set of tools and techniques and consistent information to help manage the data processing installation. For example, SLR extracts information from SMF, IMS, and CICS logs, formats selected information into tabular or graphic reports, and gives assistance in maintaining database tables.

service rate. In the system resources manager, a measure of the rate at which system resources (services) are provided to individual jobs. It is used by the installation to specify performance objectives, and used by the workload manager to track the progress of individual jobs. Service is a linear combination of processing unit, I/O, and main storage measures that can be adjusted by the installation.

shared page groups. An address space can decide to share its storage with other address spaces using a function of RSM. As soon as other address spaces use these storage areas, they can no longer be tied to only one address space. These storage areas then reside as *shared page groups* in the system. The pages of shared page groups can reside in central, expanded, or auxiliary storage.

SLA. service level agreement

SLIP. serviceability level indication processing

SLR. Service Level Reporter

SMF. System management facility

SMF buffer. A wrap-around buffer area in storage, to which RMF data gatherers write performance data, and from which the Postprocessor extracts data for reports.

speed. See *workflow*

SRB. Service request block

SRM. System resource manager

SSCH. Start subchannel

system control program (SCP). Programming that is fundamental to the operation of the system. SCPs include MVS, VM, and VSE operating systems and any other programming that is used to operate and maintain the system. Synonymous with *operating system*.

sysplex. A complex consisting of a number of coupled MVS systems.

T

tabular mode. In RMF, the mode in which Monitor III displays performance data in the form of lists. Contrast with *graphic mode*.

TCB. Task control block

threshold. The exception criteria defined on the report options screen.

throughput. A measure of the amount of work performed by a computer system over a period of time, for example, number of jobs per day.

TPNS. Teleprocessing network simulator

TSO. Time Sharing Option, see *Time Sharing Option/Extensions*

Time Sharing Option Extensions (TSO/E). In MVS, a time-sharing system accessed from a terminal that allows user access to MVS system services and interactive facilities.

U

UIC. Unreferenced interval count

uncaptured time. CPU time not allocated to a specific address space.

using. Jobs getting service from hardware resources (PROC or DEV) are *using* these resources.

V

velocity. A measure of how fast work should run when ready, without being delayed for processor or storage access. See also *execution velocity*.

VTOC. Volume table of contents

W

workflow. (1) The workflow of an address space represents how a job uses system resources and the speed at which the job moves through the system in relation to the maximum average speed at which the job could move through the system. (2) The workflow of resources indicates how efficiently users are being served.

workload. A logical group of work to be tracked, managed, and reported as a unit. Also, a logical group of service classes.

WLM. Workload Manager

WSM. Working Set Manager

X

XCF. Cross-system coupling facility

XES/CF. See *Cross-system Extended Services/Coupling Facility*.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie New York 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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.

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 OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This book documents intended Programming Interfaces that help customers to write their own RMF exit routines and to call RMF functions from their own applications.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries:

- CICS
- ESCON
- GDDM
- Hiperspace
- IBM
- IMS
- OS/390
- Processor Resource/Systems Manager
- PR/SM
- RACF
- Resource Measurement Facility
- RMF
- z/OS
- zSeries

UNIX is a registered trademark of The Open Group in the Unite· States and other countries.

Domino and Lotus are trademarks of Lotus Development Corporation.

Other company, product, and service names may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

z/OS
Resource Measurement Facility
Programmer's Guide

Publication No. SC33-7994-03

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
SC33-7994-03

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 Deutschland Entwicklung GmbH
eServer Performance Management Development
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape   **Please do not staple**   Fold and Tape

Cut or Fold
Along Line

Program Number: 5694–A01, 5655–G52

Printed in U.S.A.

SC33-7994-03

Spine information:

z/OS

z/OS V1R4 RMF Programmer’s Guide

SC33-7994-03