z/OS
2.5

Data Gatherer Programmer's Guide

IBM

> **Note**
>
> Before using this information and the product it supports, read the information in "Notices" on page 165.

Contents

Figures

Tables

x

About this document

z/OS Data Gatherer is the strategic IBM® performance measurement tool in a z/OS host environment. It measures selected areas of system activity and records the collected data in System Management Facilities (SMF) records and, in certain cases, optional VSAM data sets. Applications, such as z/OS Resource Measurement Facility (RMF), depend on the data gathered by z/OS Data Gatherer and can display that data in various formats that you can use to evaluate system performance and identify reasons for performance problems.

This document contains information and reference material to enable you to use z/OS Data Gatherer data for application programming. There are many ways to access different kinds of information, and each is described separately.

For an overview of z/OS data gathering, see *z/OS Data Gatherer User's Guide*.

Who should use this document

This information is intended for use by system programmers responsible for the development of individual, installation-specific applications in the area of gathering system measurement data. Because z/OS Data Gatherer generates information for measuring z/OS system performance, this publication assumes that the reader has extensive knowledge of the z/OS system.

z/OS Data Gatherer library

The z/OS Data Gatherer library contains the following information units:

- *z/OS Data Gatherer Programmer's Guide*
- *z/OS Data Gatherer User's Guide*

Messages issued by z/OS Data Gatherer are included in *z/OS Resource Measurement Facility Messages and Codes*.

z/OS RMF library

The z/OS RMF library contains the following information units:

- *z/OS Resource Measurement Facility Programmer's Guide*
- *z/OS Resource Measurement Facility User's Guide*
- *z/OS Resource Measurement Facility Report Analysis*
- *z/OS Resource Measurement Facility Messages and Codes* (includes z/OS Data Gatherer messages)

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to IBM Documentation (www.ibm.com/docs/en/zos).

How to read syntax diagrams

This section describes how to read syntax diagrams. It defines syntax diagram symbols, items that may be contained within the diagrams (keywords, variables, delimiters, operators, fragment references, operands) and provides syntax examples that contain these items.

Syntax diagrams pictorially display the order and parts (options and arguments) that comprise a command statement. They are read from left to right and from top to bottom, following the main path of the horizontal line.

For users accessing the Information Center using a screen reader, syntax diagrams are provided in dotted decimal format.

Symbols

The following symbols may be displayed in syntax diagrams:

Symbol
    Definition
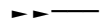
►►───
    Indicates the beginning of the syntax diagram.

───►
    Indicates that the syntax diagram is continued to the next line.

►───
    Indicates that the syntax is continued from the previous line.

───►◄
    Indicates the end of the syntax diagram.

Syntax items

Syntax diagrams contain many different items. Syntax items include:

- Keywords - a command name or any other literal information.
- Variables - variables are italicized, appear in lowercase, and represent the name of values you can supply.
- Delimiters - delimiters indicate the start or end of keywords, variables, or operators. For example, a left parenthesis is a delimiter.
- Operators - operators include add (+), subtract (-), multiply (*), divide (/), equal (=), and other mathematical operations that may need to be performed.
- Fragment references - a part of a syntax diagram, separated from the diagram to show greater detail.
- Separators - a separator separates keywords, variables or operators. For example, a comma (,) is a separator.

Note: If a syntax diagram shows a character that is not alphanumeric (for example, parentheses, periods, commas, equal signs, a blank space), enter the character as part of the syntax.

Keywords, variables, and operators may be displayed as required, optional, or default. Fragments, separators, and delimiters may be displayed as required or optional.

Item type
    Definition

Required
    Required items are displayed on the main path of the horizontal line.

Optional
    Optional items are displayed below the main path of the horizontal line.

Default
    Default items are displayed above the main path of the horizontal line.

Syntax examples

The following table provides syntax examples.

Table 1. Syntax examples

Item	Syntax example
Required item. Required items appear on the main path of the horizontal line. You must specify these items.	▶▶─ KEYWORD ── required_item ─▶◀
Required choice. A required choice (two or more items) appears in a vertical stack on the main path of the horizontal line. You must choose one of the items in the stack.	▶▶─ KEYWORD ── required_choice1 / required_choice2 ─▶◀
Optional item. Optional items appear below the main path of the horizontal line.	▶▶─ KEYWORD ── (optional_item) ─▶◀
Optional choice. An optional choice (two or more items) appears in a vertical stack below the main path of the horizontal line. You may choose one of the items in the stack.	▶▶─ KEYWORD ── (optional_choice1 / optional_choice2) ─▶◀
Default. Default items appear above the main path of the horizontal line. The remaining items (required or optional) appear on (required) or below (optional) the main path of the horizontal line. The following example displays a default with optional items.	▶▶─ KEYWORD ── default_choice1 / optional_choice2 / optional_choice3 ─▶◀
Variable. Variables appear in lowercase italics. They represent names or values.	▶▶─ KEYWORD ── *variable* ─▶◀
Repeatable item. An arrow returning to the left above the main path of the horizontal line indicates an item that can be repeated. A character within the arrow means you must separate repeated items with that character. An arrow returning to the left above a group of repeatable items indicates that one of the items can be selected, or a single item can be repeated.	▶▶─ KEYWORD ── repeatable_item ─▶◀ ▶▶─ KEYWORD ── , repeatable_item ─▶◀
Fragment. The fragment symbol indicates that a labelled group is described below the main syntax diagram. Syntax is occasionally broken into fragments if the inclusion of the fragment would overly complicate the main syntax diagram.	▶▶─ KEYWORD ── │fragment│ ─▶◀ **fragment** ▶▶── ,required_choice1 ──── / ,required_choice2 ── ,default_choice / ,optional_choice ──▶◀

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead .

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM Documentation function
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS Data Gatherer Programmer's Guide, GC31-5701-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

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

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy? OpenDocument).

Summary of changes for z/OS 2.5

Notice: This publication contains information that previously appeared in *z/OS Resource Measurement Facility (RMF) Data Gatherer Programmer's Guide*, GC27-4935-40, for z/OS 2.4.

The following content is new, changed, or no longer included in z/OS 2.5.

New

The following content is new.

August 2023 refresh

- "Monitor II REST services" on page 52 is added. (APAR OA63984)

September 2022 refresh

- Chapter 3, "Accessing data using the z/OS Data Gatherer REST services," on page 49 is added. (APAR OA62081)

June 2022 refresh

- New bit values are defined for the CPDCFLG field and new fields are added under "Channel path performance data," and a new "Channel measurements group 2 data - Extended data" section is added in "ERBCPDG3 - Channel data table" on page 90.

May 2022 refresh

- New bit values and new fields are added to support CPU topology information in "ERBCPCDB - CPC data control block" on page 84. (APAR OA62064, which also supports z/OS V2R4 and V2R3)

March 2022 refresh

- New fields are added in formerly reserved space at offsets 1172 - 1216 in "ERBASIG3 - Address space identification table" on page 66.
- New fields are added in support of transaction data for enclaves under "Resource Collection Data - Period Entry" in "ERBRCDG3 - Resource collection data" on page 130. (APAR OA62502, which also applies to z/OS V2R4 and V2R3)

Prior to March refresh

- The topic, "Obtaining SMF record data from a data set or log stream (GRBSMFR)" on page 13, is added.
- Chapter 4, "Adding Monitor I installation exits," on page 55 is added.
- ASIFLAG is added at offset 24 in "ERBASIG3 - Address space identification table" on page 66.
- CFIFLAG is added at offset 66 in "ERBCATG3 - Cache data information table" on page 74.
- CPC_Flags bits 3 and 4 are now defined in "ERBCPCDB - CPC data control block" on page 84.

- Bit 13 in the CFIENFLG field is now defined in "ERBCFIG3 - Coupling facility information table" on page 75.
- The topic, "ERBCPUDB - CPU data block" on page 94, is added.
- CPUG3_FLAG is added at offset 7 in "ERBCPUG3 - Processor data control block" on page 97.
- The topic, "ERBDDNG3 - Device data set name list table" on page 101, is added.
- GEIFLG3 is added at offset 25 in "ERBGEIG3 - General information table" on page 110.
- The topic, "ERBIQDG3 - I/O queuing performance data table" on page 117, is added.
- The topic, "ERBLOKG3 - Lock performance data table" on page 120, is added.
- In "ERBRCDG3 - Resource collection data" on page 130:
  - RCDMODE bits 2 and 3 are now defined.
  - Fields are added at offsets 296 - 308 in the Resource Collection Data - Period Entry.
  - New fields are added at offsets 132 - 168 in the Subsystem Work Manager Delays section.
- SSHVRIO, SSHIQDO, and SSHLOKO are added at offsets 324, 344, and 352, respectively, in "ERBSSHG3 - MINTIME set of samples header" on page 141.
- The topic, "ERBVRIG3 - VSAM RLS information data table" on page 150, is added.
- ZFXFlag is added at offset 52 in "ERBZFXG3 - zFS performance data table" on page 156.

Changed

The following content is changed.

February 2023 refresh

- Reserved fields are documented at the end of the "Home LPAR section" in "ERBCPCDB - CPC data control block" on page 84.
- The name and description of the field at offset 60, as well as the name and description of the field at offset 6 under "CPU extension for each processor," are updated in "ERBCPUDB - CPU data block" on page 94.
- The length of the reserved field at offset 164 (A4) and the descriptions of the CPUG3_PARK_IFA and CPUG3_PARK_SUP fields are updated in "ERBCPUG3 - Processor data control block" on page 97.
- The descriptions of the RCDIFAT and RCDIFCT fields are updated in "ERBRCDG3 - Resource collection data" on page 130.

August 2022 refresh

- Bit 10 in the CPUG3_STATUS field is now defined in "ERBCPUG3 - Processor data control block" on page 97. (APAR OA63190, which also applies to z/OS 2.4.)

June 2022 refresh

- The description of the IQDLMCUs field is updated in "ERBIQDG3 - I/O queuing performance data table" on page 117.

May 2022 refresh

- Bits 8 and 10 in the ASIMSTS field are updated in "ERBASIG3 - Address space identification table" on page 66. (APAR OA60660, which also applies to z/OS V2R4)
- The lists of values are updated in the descriptions of the APType and MapType fields in "ERBCRYG3 - Cryptographic hardware data table" on page 99. (APAR OA59874, which also applies to z/OS V2R4 and V2R3)
- Additional flag values are defined for the CPUFLAGS field in "ERBCPUDB - CPU data block" on page 94. (APAR OA62064, which also supports z/OS V2R4 and V2R3)

Prior to May 2022 refresh

- The terms *RMF Data Gatherer* and *RMF data gathering* have been updated to *z/OS Data Gatherer* and *z/OS data gathering* throughout the text.
- The list of measurement tables or records is updated in "Data set content" on page 62.
- The meaning of return code 16, reason code 92 is updated in "Return codes and reason codes" on page 35.
- Obsolete fields at decimal offsets 44, 48, 128, and 194 have been changed to "reserved" in "ERBASIG3 - Address space identification table" on page 66.
- Obsolete fields at decimal offsets 96, 104, 112, 348, 352, 356, and 360 have been changed to "reserved" in "ERBGEIG3 - General information table" on page 110.

Deleted

The following content was deleted.

- Return codes 116 and 120 are removed from the table in "Return codes" on page 11.
- The chapter, "Adding Monitor III user exits," has been removed.

Summary of changes for Version 2 Release 4

> **Notice:** This publication contains information that previously appeared in *z/OS RMF Programmer's Guide*, SC34-2667.
>
> This publication presents programming information specific to RMF data gathering. For programming information specific to RMF reporting, see *z/OS RMF Reporter Programmer's Guide*, GC27-4937.

New

The following information is added in this publication:

Prior to February 2021 refresh

- Entries have been added to the CFISTRUS Table Entry Section in "ERBCFIG3 - Coupling facility information table" on page 75. (APAR OA58729)
- Entries have been added to the list in "Data set content" on page 62.
- Entries have been added to the table in "ERBSSHG3 - MINTIME set of samples header" on page 141.
- The following tables have new fields and updated field descriptions to support restricted use common service area (RUCSA):
  - "ERBASIG3 - Address space identification table" on page 66
  - "ERBCSRG3 - Common storage remaining table" on page 100
  - "ERBGEIG3 - General information table" on page 110
- Added a table in "ERBXCFG3 - XCF Activity data table" on page 153.
- Information about Extended Asynchronous Data Mover (EADM) has been added in "ERBSCMG3 - Extended Asynchronous Data Mover (EADM) data table" on page 138.

Changed

The following information is changed in this publication:

April 2021 refresh

- The CPC_BoostType field was incorrectly displayed under the CPC Logical Processor Section; it is now correctly displayed under the CPC LPAR Section in "ERBCPCDB - CPC data control block" on page 84.

February 2021 refresh

- The service routine and meaning are updated for return code 16, reason code 53 for RMF Sysplex Data Services in "Return codes and reason codes" on page 35.

Prior to February 2021 refresh

- Updated tables in "ERBCPCDB - CPC data control block" on page 84.

Deleted

Prior to February 2021 refresh

- Information about diagnosing problems in RMF has moved to *z/OS RMF Messages and Codes*, SC34-2666.

Chapter 1. SMF records

The topics that follow cover the following items:

- Summary of all SMF record types generated by z/OS Data Gatherer
- 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, z/OS Data Gatherer 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 z/OS Data Gatherer is described in *z/OS MVS System Management Facilities (SMF)*.

RMF does not generate reports from SMF record type 72 subtype 4; however, these records are available for user-written reports.

You use the SMFBUF option to define the SMF record types and subtypes to be written, which you can specify in the following ways:

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

The record types and the corresponding 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 3 – Workload activity
  - Subtype 4 – Storage data
  - Subtype 5 – Serialization delay
- 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
  - Subtype 8 – Enterprise disk system statistics

- Subtype 9 – PCIE based function activity
- Subtype 10 – EADM activity
- 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 9 – Device activity
  - Subtype 11 – Paging data set activity
  - Subtype 12 – Channel path activity
  - Subtype 14 – I/O queuing activity
  - Subtype 15 – IRLM long locks
- Record type 104 serves as a container for performance measurement data collected by RMF XP from non z/OS platforms:
  - Subtype 1-12 – Performance data from AIX® on System p
  - Subtype 20-31 – Performance data from Linux® on System x
  - Subtype 40-53 – Performance data from Linux on IBM Z

You can find details about which monitor is writing which SMF records in *z/OS Data Gatherer User's Guide*.

SMF record format

Depending on the feedback options you select, z/OS Data Gatherer can write the SMF records to the SMF data set, and RMF can use the data in the record to generate a printed report. Regardless of the options you select, the format of the SMF record is the same.

Each SMF record that z/OS Data Gatherer 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 data gatherer 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 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 z/OS Data Gatherer 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 on page 3 shows the general format of the SMF records that z/OS Data Gatherer 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.
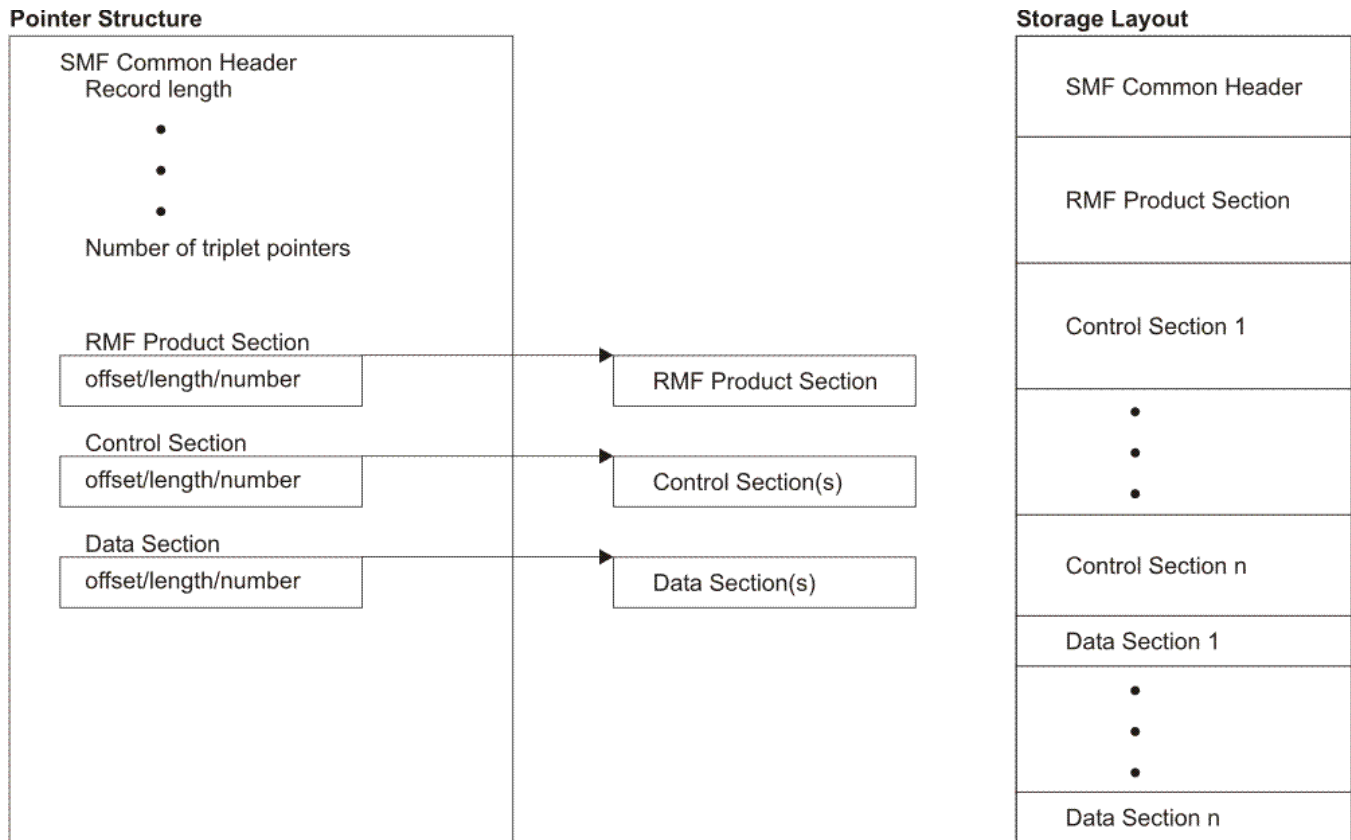
Figure 1. SMF record format

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 z/OS Data Gatherer 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 z/OS Data Gatherer 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.

Archived performance data

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

Data gatherer version numbers

Applications, such as the RMF Postprocessor, can use the z/OS Data Gatherer version number of each SMF record for its own purposes. The version number appears in the SMF*xx*MFV field, where *xx* is the record number. For example, the field may contain one of the following values:

- X'780F' for an SMF record produced by z/OS V1R13 RMF
- X'790F' for an SMF record produced by z/OS V2R1 RMF
- X'792F' for an SMF record produced by z/OS V2R2 RMF
- X'794' for an SMF record produced by z/OS V2R3 RMF
- X'796' for an SMF record produced by z/OS V2R4 Data Gatherer
- X'797' for an SMF record produced by z/OS V2R5 Data Gatherer

When the version number indicates that the record was produced by an earlier version or release, the Postprocessor converts the record to the current data gatherer 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 the data gatherer 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 the original record version number is preserved.

These differences will also be reflected accordingly in the RMF reports.

Printing SMF records

You might occasionally find it necessary to print the SMF records that z/OS Data Gatherer 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, can print all SMF records in dump format.
- The z/OS Data Gatherer utility program, ERBSCAN, running under ISPF can format all data gatherer records in record-type-specific sections.

Using the IDCAMS utility

A sample of the JCL needed to print SMF records follows. The first job step (SELECT) invokes the IFASMFDP dump program to limit the amount of output to the record types or time frames that you want. If you want to print the entire data set, use only the second job step (PRINT), defining 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, 2021.

```
//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(20210403,20210403)
/*
//PRINT     EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=A
//RMFREC    DD    DSN=&&RMFREC,DISP=(OLD,PASS)
//SYSIN     DD    *
  PRINT   INFILE(RMFREC)
/*
```

Using IFASMFDP - the SMF data set dump program in *z/OS MVS System Management Facilities (SMF)* contains more information about the IFASMFDP dump program. *z/OS DFSMS Access Method Services Commands* contains more information about the IDCAMS utility.

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 2 on page 5 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      04/28/21    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 2. Dump Format of SMF Record

Using the ERBSCAN utility

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

```
ERBSCAN data-set-name
```

```
VIEW        SYS21119.T130441.RA000.BTEU.R0500089            Columns 00001 00072
Command ===>                                                Scroll ===> HALF
****** *************************** Top of Data ****************************
000001 1z/OS V2R5 Data Gatherer ERBMFSCN Version 9 (30 Apr 2012) - 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: 2021 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 2021.117 14:19:01                                  VSP
000015       2 074.002 15064 2021.094 19:14:00 2021.094 18:59:00 15:00:000 OS0
000016       3 074.003   412 2021.094 19:14:00 2021.094 18:59:00 15:00:000 OS0
000017       4 073.001 18680 2021.094 19:14:00 2021.094 18:59:00 15:00:090 OS0
000018       5 074.004  1912 2021.094 19:14:00 2021.094 18:59:00 15:00:000 OS0
000019       6 074.004  3816 2021.094 19:14:00 2021.094 18:59:00 15:00:000 OS0
000020       7 078.003 29976 2021.094 19:14:00 2021.094 18:59:00 15:00:090 OS0
000021       8 074.001 15520 2021.094 19:14:03 2021.094 18:59:00 15:00:090 OS0
000022       9 074.001 32604 2021.094 19:14:03 2021.094 18:59:00 15:00:090 OS0
```

Figure 3. ERBSCAN - Display SMF record list

The ERBSHOW function is part of ERBSCAN to format a specific record. You call ERBSHOW and specify a record number, as in the following 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        SYS21119.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        : 21.094
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 4. ERBSHOW - Display SMF record header

Scrolling forward leads to the different sections of the record.

```
VIEW        SYS21119.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 5. ERBSHOW - Display Device Data Section

Obtaining Monitor II SMF record data directly (ERBSMFI)

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

Description

To use the data interface service, invoke the ERBSMFI module with the registers and parameters described in .

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 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 in *z/OS Resource Measurement Facility Programmer's Guide*.

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 Monitor II data interface service contains nine fullword pointers, which contain the addresses of the following parameters:

Parameter 1
> Fullword. Request type:
>
> **1**
>> Parameter list contains 7 parameters
>
> **2**
>> Parameter list contains 8 parameters
>
> **3**
>> Parameter list contains 9 parameters
>
> **4**
>> Parameter list contains 10 parameters
>
> **5**
>> Parameter list contains 11 parameters
>
> **6**
>> Parameter list contains 11 parameters and if SMF 79 record subtype 9 is requested and ERBSMFI detects more than 65535 active devices, return code 64 is provided in register 15.

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 10.

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, as partial data will not be returned. For other SMF record type 79 subtypes, partial data is returned if the buffer is not long enough.

Parameter 6
> Fullword. Returns the system CPU utilization of standard CPs.

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)

9
DEV(NUMBER(0:FFFF))

11
PGSP

12
CHANNEL

14
IOQUEUE(DASD)

This parameter allows you to pass certain RMF report options to the Monitor II data gatherer when parameter 1 contains the request type 2 or higher. See Snapshot reporting with Monitor II in *z/OS Resource Measurement Facility User's Guide* for other options. Use the display-session syntax described there.

Parameter 9
Fullword. Returns the MVS™/SRM CPU utilization of standard CPs.

This parameter is returned for request type **≥3**.

Parameter 10
Fullword. Returns the system CPU utilization of ZAAPs.

This parameter is returned for request type **≥4**.

Parameter 11
Fullword. Returns the system CPU utilization of ZIIPs.

This parameter is returned for request type **≥5**.

Example:

To generate data for the Monitor II Device Activity report for those devices that are physically configured to subchannel set 0 and have device addresses in the range of 0000 to 2FFF, you would have to issue the command:

```
DEV NUM(00000:02FFF)
```

You can specify this command with the following parameter list:

Addr(P1)	5	Parameter list contains 11 parameters
Addr(P2)	79	SMF type 79
Addr(P3)	9	Subtype 9 (Device Activity)
Addr(P4)	Output buffer	
Addr(P5)	L(Output buffer)	
Addr(P6)		
Addr(P7)		L(options field)
Addr(P8)	16 NUM(00000:02FFF)	Report options
Addr(P9)		
Addr(P10)		
Addr(P11)		
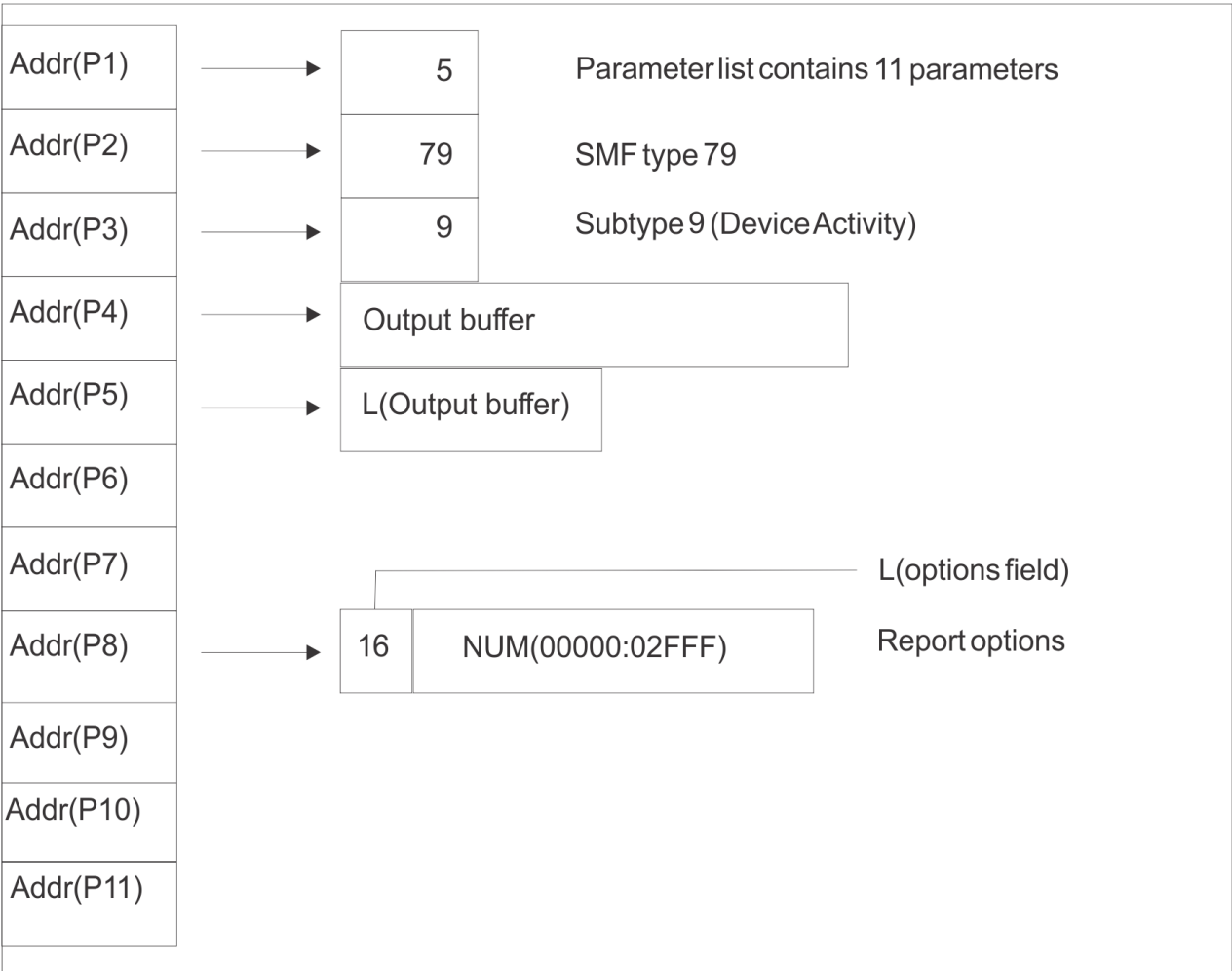
Figure 6. Parameter list for a command to obtain SMF record data

Output

The following are considerations for the output parameters:

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 z/OS Data Gatherer 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 standard CP utilization percent as a binary fullword in the area provided. If z/OS Data Gatherer cannot determine the CPU utilization percent on a PR/SM system because the Monitor I CPU report is not active, it 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).

Parameter 10
>   Contains the current average ZAAP utilization percent as a binary fullword in the area provided. If z/OS Data Gatherer cannot determine the ZAAP utilization percent on a PR/SM system because the Monitor I CPU report is not active, it returns a value of -1 (FFFFFFFF).

Parameter 11
>   Same as Parameter 10, but for ZIIPs.

Return codes

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

Table 2. Return codes for the Monitor II Data Interface Service (ERBSMFI)

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.
64	ERBSMFI can not process data for more than 65535 devices. Specify a device number range in report command DEV which does not encompass more than 65535 devices, e.g. 00000:0FFFF.
100	Input record type or subtype is not valid.

Table 2. Return codes for the Monitor II Data Interface Service (ERBSMFI) (continued)

Return code (Decimal)	Description
104	No data is returned; SMF record buffer is too short.
108	Request type is not known.
112	ESTAE routine had control.
124	The user is not authorized to access Monitor II data.

Coded example

The following Assembler code example calls the 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
```

Obtaining SMF record data from a data set or log stream (GRBSMFR)

You can use the GRBSMFR service to retrieve SMF records from an SMF data set or log stream and, optionally, convert records to a different release or service level, or reassemble broken records, or both.

Description

SMF post-processing applications must be able to retrieve into storage SMF records that reside in an SMF data set or SMF log stream. Such applications often face the problem that SMF records that were written on another z/OS release or service level must either be up-converted to a higher service level or down-converted to an older service level. z/OS Data Gatherer provides a conversion method in the GRBSMFR service that adapts SMF data section lengths and data fields in such a way that the calling program can read and understand the converted data. If the SMF records were physically broken by the data gatherer, the GRBSMFR service can be used to reassemble these records into a single, large logical SMF record.

The GRBSMFR service can be invoked to request three different functions: OPEN, GET, or CLOSE. Each function requires the specification of a set of parameters, as described in . The parameters must be coded in the specified order and the values assigned to the parameters must be in the required format.

Environment

The requirements for the caller are:

Minimum authorization:	Problem state. Any PSW key.
Dispatchable unit mode:	Task mode.
Cross-memory mode:	Any PASN, any HASN, any SASN.

AMODE:	64-bit. Code SYSSTATE AMODE64=YES before invoking this service.
ASC mode:	Primary.
Interrupt status:	Enabled for I/O and external interrupts.
Locks:	No locks held.

Syntax

The GRBSMFR service has the following calling syntax:

Main diagram:
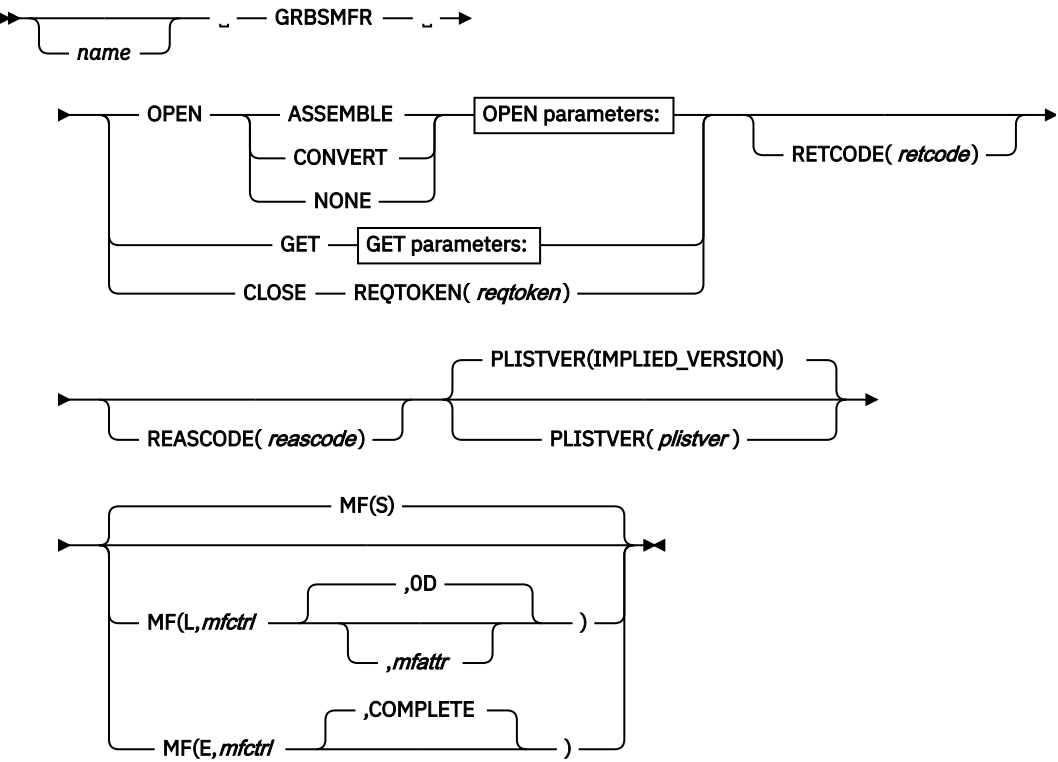
OPEN parameters:

GET parameters:

Parameters

The GRBSMFR service has the following parameters:

name
> An optional symbol, starting in column 1, that is the name on the GRBSMFR macro invocation. The name must conform to the rules for an ordinary assembler language symbol.

OPEN, GET, and CLOSE is a set of mutually exclusive keywords. This set is required; only one keyword must be specified.

OPEN
> The OPEN function examines the SMF data set or log stream and returns information that can be used as input for calling the GET function. The OPEN request must be invoked before the first GET request.

ASSEMBLE, CONVERT, and NONE is a set of mutually exclusive keywords. This set is required; only one keyword must be specified.

ASSEMBLE
Specifies that SMF records written on a different z/OS release or service level must be converted to the SMF record level that is supported by this service. In addition, all broken SMF records are reassembled into one large logical SMF record.

CONVERT
Specifies that SMF records written on a different z/OS release or service level must be converted to the SMF record level that is supported by this service. No SMF record reassembly is done.

NONE
Specifies that SMF records must neither be converted to another SMF record level nor reassembled into a large logical SMF record.

REQTOKEN(*reqtoken*)
Specifies the name of a required 4-byte signed integer output variable in which a request token is returned by the OPEN function. The request token must be used for subsequent GET requests and the final CLOSE request.

DDNAME(*ddname*)
Specifies the name of a required 8-byte character input variable which specifies the DDNAME of the SMF data set or log stream that contains the SMF records to be retrieved. If the DDNAME represents an SMF log stream, the DD statement must be coded as follows:

```
DD DSN=log-stream-name,DISP=SHR,DCB=(RECFM=VB,BLKSIZE=32760),
        SUBSYS=(LOGR,IFASEXIT,'FROM=(yyyy/ddd,hh:mm),TO=(yyyy/ddd,hh:mm),LOCAL')
```

FILTLIST(*filtlist*)
Specifies the name of a required 40-byte character input variable which specifies the filter criteria to be applied to the input SMF records. The input SMF records are provided in the SMF data set or log stream specified by the DDNAME parameter.

Specify one or more of the following filters in *filtlist*:

Bytes	Filter	Description
0 - 1	Version	A 2-byte unsigned integer value that specifies the version of the filter list. Must be set to 1.
2 - 3	Length	A 2-byte unsigned integer value that specifies the length of the filter list. Must be set to 40.
4 - 5	Type	A 2-byte unsigned integer value that specifies the SMF type of the records to be processed. Specify binary zeros if you do not want to apply this filter to select input SMF records.
6 - 7	Subtype	A 2-byte unsigned integer value that specifies the SMF subtype of the records to be processed. Specify binary zeros if you do not want to apply this filter to select input SMF records.
8 - 11	SMFID	A 4-byte character value that specifies the SMF ID of the z/OS system that has written the SMF records to be processed. Specify blanks if you do not want to apply this filter to select input SMF records.
12 - 25	Start time	A 14-byte character value in the format *yyyymmddhhmmss* that specifies the beginning of the time interval for which information is requested. Specify blanks if you do not want to apply this filter to select input SMF records.
26 - 39	End time	A 14-byte character value in the format *yyyymmddhhmmss* that specifies the end of the time interval for which information is requested. Specify blanks if you do not want to apply this filter to select input SMF records.

ANSAREAP(*ansareap*)
Specifies the name of a required 8-byte character output value that specifies the 64-bit address of the area to which the GRBSMFR service returns the information requested by the caller. The area must be in the caller's primary address space.

The answer area contains the SMF data set lookup table (DSLT), which consists of a header and one or more data sections. Each data section provides the calling program with information about one specific SMF record in the SMF data set or log stream. Only SMF records that match the specified selection criteria are described in DSLT data sections.

The DSLT header has the following format:

Bytes	Data	Description
0 - 3	Acronym	The 4-character acronym, DSLT, of the data set lookup table.
4 - 5	Version	The version of the data set lookup table.
6 - 7	Header length	The length of the header of the data set lookup table.
8 - 11	Total length	The total length of the data set lookup table, including all data sections.
12 - 15	Number of data sections	The total number of data sections in the data set lookup table.
16 - 17	Data section length	The length of one data section.
18 - 19	Reserved	Reserved space that contains the value X'0000'.
20 - 23	Number of SMF record sets	The total number of SMF record sets in the data set lookup table.
24 - 31	Request token	The request token returned by the OPEN function that connects the OPEN, GET, and CLOSE requests.

Each DSLT data section has the following format:

Bytes	Data	Description
0 - 3	DSLT record ID	The identifier of the SMF record in the data set lookup table.
4 - 7	SMF record ID	The identifier of the SMF record in the SMF data set or log stream.
8 - 11	Record set token	The identifier of the SMF record set which consists of either one unbroken SMF record or multiple broken records that are reassembled into one large logical SMF record.
12 - 13	Broken record number	The sequence number of the broken SMF record set identified by the record set number.
14 - 15	Total broken records	The number of broken records that belong to the SMF record set identified by the record set number.
16 - 17	SMF type	The SMF record type, as was specified by the type filter in the FILTLIST parameter.
18 - 19	SMF subtype	The SMF record subtype, as was specified by the subtype filter in the FILTLIST parameter.
20 - 23	SMF ID	The SMF ID of the z/OS system that wrote the SMF record, as was specified by the SMFID filter in the FILTLIST parameter.
24 - 27	Interval start date	The start date of the SMF interval in the form X'$0cyyddd$F', as was specified in the form $yyyymmdd$ by the start time filter in the FILTLIST parameter.
28 - 31	Interval start time	The start time of the SMF interval in the form X'$0hhmmss$F', as was specified in the form $hhmmss$ in the start time filter in the FILTLIST parameter.
32 - 35	Interval duration	The length of the SMF interval, in the form X'$mmssttt$F'.
36	DSLT flags	DSLT flag byte.

Bit

    Meaning when set

0

    The SMF subtype in bytes 18 - 19 of the DSLT is valid.

1

    The interval start date, start time, and duration in bytes 24 - 35 are valid.

2 - 7

    Reserved.

Bytes	Data	Description
37	Unused	Reserved.
38 - 39	SMF record length	The length of the SMF record.
40 - 43	SMF record move time	The time since midnight, in hundredths of a second, when the record was moved into the SMF buffer. In record types 2 and 3, this is the time the record was moved to the dump data set.
44 - 47	SMF record move date	The date when the record was moved into the SMF buffer, in the form X'0*cyyddd*F'. In record types 2 and 3, this is the date the record was moved into the dump data set.

ANSAREAL(*ansareal*)
> The name of a required, 4-byte, unsigned integer input variable that specifies the length of the allocated answer area. If you do not provide enough space, the GRBSMFR service returns in this parameter the length needed for the complete data. The length value returned on OPEN requests is the size of the SMF data set lookup table.

GET
> The GET function retrieves either broken, unbroken, or reassembled SMF records from the SMF data set or log stream specified in the DDNAME parameter. If an SMF record is at a different service level than this service, the SMF records are converted to the current service level before they are reassembled.
>
> The GET request must be invoked after the OPEN request.

REQTOKEN(*reqtoken*)
> Specifies the name of a required, 4-byte, signed integer input variable which contains the request token that was returned by the OPEN function.

RSTOKEN(*rstoken*)
> Specifies the name of a required, 4-byte, signed integer input variable which specifies the record set token that identifies the SMF record set that is to be processed by the GET function. Valid record set tokens are returned in the answer area of the OPEN request in bytes 8 - 11 of each data section.

ANSAREAP(*ansareap*)
> Specifies the name of a required, 8-byte, character output variable that specifies the 64-bit address of the area to which the GRBSMFR service returns the requested information. The area must be in the caller's primary address space. The area contains a 4-byte prefix area that contains the SMF record length followed by exactly one SMF record that is either an unbroken or reassembled SMF record. The first 2 bytes of the SMF record provide the length of the SMF record if it is not reassembled. If the returned SMF record was reassembled, bytes 0 and 1 of the SMF record are set to zero.

ANSAREAL(*ansareal*)
> Specifies the name of a required, 4-byte, unsigned input variable that specifies the length of the allocated answer area. If you do not provide enough space, the GRBSMFR service returns in this parameter the length needed for the complete data. The value returned on GET requests is the estimated length of the area including the prefix area and the reassembled SMF record. The size can only be estimated to avoid repositioning actions on the SMF data set due to unsuccessful read operations.

CLOSE
> The CLOSE function closes the SMF data set or log stream and cleans up internal work areas. The CLOSE request must be invoked after the last invocation of the GET function for a specific data set or log stream.

REQTOKEN(*reqtoken*)
> Specifies the name of a required, 4-byte, signed input variable which contains the request token that was returned by the OPEN function.

RETCODE(*retcode*)
Specifies the name of an optional, 4-byte, signed output variable into which the return code is to be copied from GR 15.

REASCODE(*reascode*)
Specifies the name of an optional, 4-byte, signed output variable to contain the reason code.

PLISTVER(*plistver*|IMPLIED_VERSION)
Specifies an optional, 1-byte, decimal value in the range 0 - 0 that specifies the macro version of the GRBSMFR service. PLISTVER is the only keyword allowed on the list form of MF and determines which parameter list is generated. Note that you may specify MAX instead of a number, and the parameter list will be of the largest size currently supported. This size may grow from release to release (thus possibly affecting the amount of storage needed by your program). If your program can tolerate this, IBM recommends that you always specify MAX when creating the list-form parameter list, as this ensures that the list-form parameter list is always long enough to hold whatever parameters might be specified on the execute form.

Default: IMPLIED_VERSION

When PLISTVER is not specified, the default is the lowest version which allows all of the parameters specified on the invocation to be processed.

MF(S)
MF(L,*mfctrl*[,[*mfattr*|0D]])
MF(E,*mfctrl*[,COMPLETE])
An optional keyword input parameter that specifies the macro form.

MF(S)
Specifies the standard form of the macro. The "S" form generates code to put the parameters into the parameter list and invoke the desired service. Full checking for required macro keywords is done along with supplying defaults for omitted optional parameters. The ?EPILOG macro must be invoked at the end of the module.

MF(L,*mfctrl*[,[*mfattr*|0D]])
Specifies the list form of the macro. The "L" form defines an area to be used for the parameter list. Only the PLISTVER keyword may be specified on the invocation. All other parameters are flagged as errors. If PLISTVER is not specified, the original parameter list definition is used.

,*mfctrl*
A required input, which is the name of a storage area for the parameter list.

,*mfattr*|0D
An optional 60-character input string that varies from 1 to 60 characters. Use it to force boundary alignment of the parameter list. Use only 0F or 0D.

Default: ,0D which forces the parameter list to a doubleword boundary.

MF(E,*mfctrl*[,COMPLETE])
Specifies the execute form of the macro. The "E" form generates code to put the parameters into the parameter list specified by *mfctrl* and provides full syntax checking with default setting.

,*mfctrl*
A required input, which is the name of a storage area for the parameter list.

,COMPLETE
An optional keyword input that specifies the degree of macro parameter syntax checking.

Default: COMPLETE

Default: MF(S)

Return and reason codes

The return and reason codes are grouped into classes that indicate 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.

Warning (RC = 4)
> The returned data may be inconsistent.

Error (RC = 8)
> No data was returned, for example, because invalid parameters were specified by the caller.

Severe error (RC = 16)
> A problem has been detected within SMF processing.

Table 3. Return and reason codes for the GRBSMFR service (as declared in the GRBSMFRC macro)		
Return code	**Reason code**	**Meaning and action**
0	0	**Meaning:** The operation was successful. The answer area contains the requested data. **Action:** Continue normal program execution.
4	0	**Meaning:** No SMF record data was found that matches the filter criteria specified in the filter list. **Action:** Specify other filter criteria in the FILTLIST parameter or continue normal program execution.
4	4	**Meaning:** At least one broken SMF record was detected in an SMF record set that belongs to a different time interval or was collected on a different system than the other records in the SMF record set. The inconsistent SMF record is ignored. **Action:** Ensure that only consistent SMF record sets are passed to the service.
4	8	**Meaning:** At least one SMF record was not converted to the service level supported by z/OS Data Gatherer. **Action:** The service provides the calling program with the unconverted SMF record in the answer area. Contact your system administrator and request the latest z/OS Data Gatherer service to be installed.
4	12	**Meaning:** At least one broken SMF record was found that cannot be reassembled by the service. **Action:** The service provides the calling program with the broken SMF record in the answer area.
4	16	**Meaning:** At least one incomplete SMF record set was found that cannot be reassembled by the service. **Action:** The service provides a DSLT entry in the answer area. The DSLT is supposed to belong to the incomplete SMF record set. Use the DSLT to determine which SMF records are affected and remove those records from the SMF data set or exclude them by using another filter in the FILTLIST parameter.
4	20	**Meaning:** At least one SMF record was converted to the version or SMF record level that is supported by GRBSMFR. **Action:** The service provides the calling program with the converted SMF record in the answer area.
8	4	**Meaning:** The answer area provided by the calling program was too small for the service to return all the requested information. **Action:** Use the required answer area length returned by the OPEN or GET request to provide an answer area large enough to contain the reassembled SMF record.
8	8	**Meaning:** The start or end time in the FILTLIST parameter was specified in an invalid format. **Action:** Specify start and end time in the format *yyyymmddhhmmss* and rerun the program.
8	12	**Meaning:** The specified request token is invalid. **Action:** Specify the request token that was passed back on completion of the OPEN request.

Return code	Reason code	Meaning and action
		Table 3. Return and reason codes for the GRBSMFR service (as declared in the GRBSMFRC macro) (continued)
8	16	**Meaning:** The specified record set token is invalid. **Action:** Use the DSLT to determine a valid SMF record set ID and specify the ID as the SMF record set token.
8	20	**Meaning:** The specified request type is invalid. **Action:** Specify a request type of OPEN, GET, or CLOSE.
8	24	**Meaning:** The calling program is not in task mode. **Action:** Rerun your program in the correct mode.
8	28	**Meaning:** The calling program is not enabled. **Action:** Rerun your program in the correct mode.
8	32	**Meaning:** The calling program is not unlocked. **Action:** Rerun your program in the correct mode.
16	4	**Meaning:** The service was unable to access the SMF data set or log stream.
16	8	**Meaning:** Internal error. The service was unable to establish a recovery environment.
16	12	**Meaning:** Unexpected error. The error recovery routine of the service had control.
16	16	**Meaning:** Unexpected internal error.
16	20	**Meaning:** Unexpected error. Number of triplets in SMF record is higher than supported by this service.

Chapter 2. z/OS Data Gatherer sysplex data services

The information in this chapter describes callable services (available for 31-bit and 64-bit environments) that enable you to access sysplex data:

- ERBDSQRY/ERBDSQ64 - Query Available Sysplex SMF Data Service
- ERBDSREC/ERBDSR64 - Request Sysplex SMF Record Data Service
- ERB2XDGS/ERB2XD64 - Monitor II Sysplex Data Gathering Service
- ERB3XDRS/ERB3XD64 - Monitor III Sysplex Data Retrieval Service

This chapter describes the CALL statements that invoke z/OS Data Gatherer 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 sysplex data services, you issue CALLs that invoke the appropriate data service program. All of them are available as a set of APIs for the 31-bit environment, as well as for the 64-bit environment (see). 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 supplied SAMPLIB contains three sample programs (written in C) that invoke these services (only in 31-bit mode):

- 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 supplied exit ERB2XSMF which returns the complete SMF type 79 records, or one might write their own exit routine for a data reduction.

For ERB3XDRS (ERBDSMP3 sample program), the 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
```

Note: For information about the RACF® definitions needed to allow access to the sysplex data services, see Controlling access to data for the sysplex data services in *z/OS Data Gatherer User's Guide*.

How to call sysplex data services in 64-bit mode

With the introduction of the z/Architecture®, applications can run in 64-bit addressing mode. The sysplex data services mentioned above can also be called by 64-bit callers using alternate entry points as described in Table 4 on page 22:

Table 4. Sysplex data services	
31-bit API	**64-bit API**
ERBDSQRY	ERBDSQ64
ERBDSREC	ERBDSR64
ERB2XDGS	ERB2XD64
ERB3XDRS	ERB3XD64

The parameters for the 64-bit API are identical to those for the 31-bit API. The 64-bit APIs may be called with parameters located above or below the 2 GB bar, except the answer area which must be located below the 2 GB bar.

Note: Information provided for the 31-bit APIs is also valid for the 64-bit APIs, even though not explicitly mentioned. Exceptions from that rule are indicated where required.

ERBDSQRY - Query available sysplex SMF data service

Call ERBDSQRY to request a directory of SMF record data available in the 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 5. ERBDSQRY Service	
Service call statement	**Parameters**
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 the service 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, the service 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 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 z/OS Data Gatherer 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 z/OS Data Gatherer measurement interval (also part of the RMF product section of the SMF record).

The directory entries returned by ERBDSQRY contain SMF record header, z/OS Data Gatherer measurement interval information, plus a record token.

See .

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

Figure 7. Format of the start time

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 data gatherer 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 data gatherer 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 - 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, the service 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 35.

,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 35.

ERBDSREC - Request sysplex SMF record data service

Call ERBDSREC to request SMF record data from the data gatherer 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 6. ERBDSREC Service

Service call statement	Parameters
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 the service 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, the service 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.

#tokens	token1	token2	…

,time_out
Specifies a time interval in seconds. If the time interval expires during the processing of the service, the service 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 35.

,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 35.

ERB2XDGS - Monitor II sysplex data gathering service

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 7. ERB2XDGS Service

Service call statement	Parameters
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 the service 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, the service 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.

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 ...
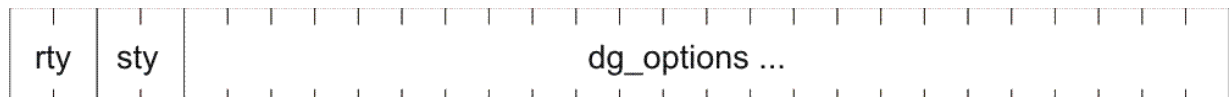
where:

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.

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.

Note: If device performance data for DASDs is requested in a IBM zHyperWrite environment (sty = 09), it is recommended to invoke ERB2XDGS multiple times with a device number range specified in dg_options that does not encompass more than 65535 devices, e.g. 00000:0FFFF. If ERB2XDGS requests performance data for more than 65535 devices (for instance, dg_options = DASD), only data for a maximum of 65535 devices are passed back in the returned SMF type 79 subtype 9 record. This condition is indicated by return code 64 provided in field SRC at offset x'2C' in the ERB2XDGS/ ERB2XD64 data section header for each SMF system ID.

,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 z/OS Data Gatherer on each system from which data is requested. After the Monitor II data has been retrieved by the data gatherer, this exit may copy selected areas from the data to the answer area provided by the data gatherer. These data areas are then combined into the answer area provided by the caller of ERB2XDGS.

The data reduction ERB2XSMF exit routine 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, the service 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 the service 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.

For details see "Return codes and reason codes" on page 35.

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

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

For details see "Return codes and reason codes" on page 35.

ERB2XDGS data reduction exit routines

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 as shown, with the parameters in the specified order.

Table 8. ERB2XDGS Exit Routine	
Service call statement	**Parameters**
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. The service 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 that the caller provided to ERB2XDGS, including the common header and data headers created by z/OS Data Gatherer.

  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.

ERB3XDRS - Monitor III sysplex data retrieval service

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 9. ERB3XDRS Service

Service call statement	Parameters
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 the service 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, the service indicates 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.
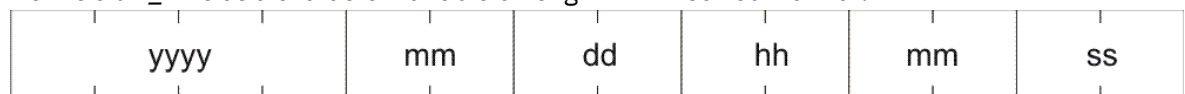
yyyy	mm	dd	hh	mm	ss

　　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 by using the ERB3RDEC service.

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 on each system from which data is requested. After the set-of-samples data has been retrieved, this exit may copy selected areas from the set-of-samples to the answer area provided by the data gatherer. These data areas are then combined into the answer area provided by the caller of ERB3XDRS.

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, the service 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 result in the 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.

For details see "Return codes and reason codes" on page 35.

,reason_code
When ERB3XDRS 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 35.

ERB3XDRS data reduction exit routines

The exit routine specified in the **exit_name** parameter of the ERB3XDRS service is invoked on each system to which the ERB3XDRS 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 as shown, with the parameters in the specified order.

Table 10. ERB3XDRS exit routine	
Service call statement	**Parameters**
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 in the *answer_area_addr* parameter. z/OS Data Gatherer provides an answer area of 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 the data gatherer.

  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			V	*		DRC		DLN	
FSS		LSS			*				
				*					
FAV			LAV						
				*					
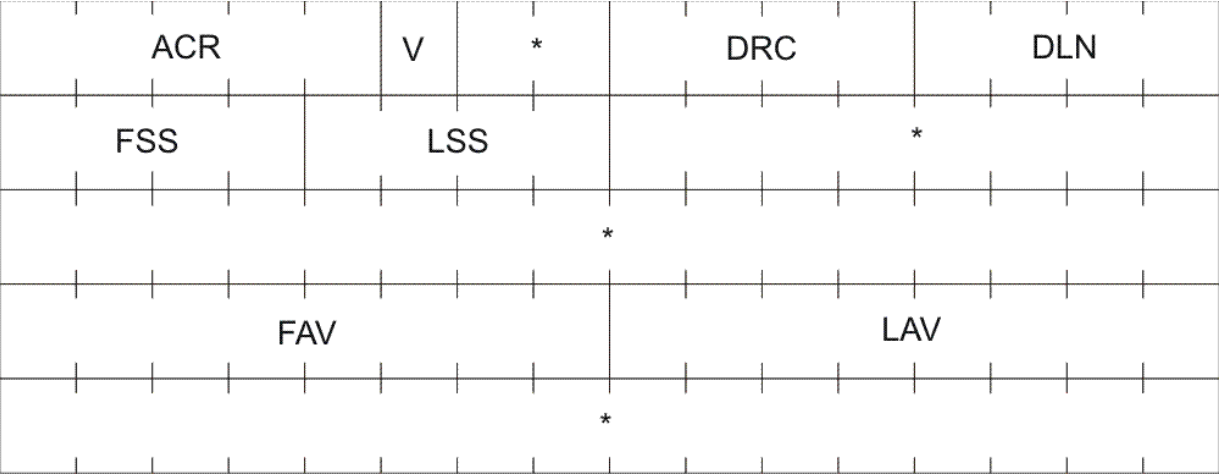
ACR

(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 a 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 an unsigned integer variable of length 4.

Return codes and reason codes

When the sysplex data services 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 .

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 and ERB3XD64 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 data gatherer 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 and ERBDSQ64

R
> ERBDSREC and ERBDSR64

2
> ERB2XDGS and ERB2XD64

3
> ERB3XDRS and ERB3XD64

Table 11. 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.

Table 11. Sysplex data services return and reason codes (SMF services) (continued)

Return code	Reason code	Service routine	Meaning
			Action
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 Monitor III 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 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 - The 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.

Table 11. Sysplex data services return and reason codes (SMF services) (continued)			
Return code	Reason code	Service routine	Meaning
			Action
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.
12	6	-,-,2,-	**Meaning:** Error - No 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. Pre-allocated data sets unusable (detected at start of retrieval).
12	19	-,-,-,3	**Meaning:** Error - no data returned by ERB3XDRS. Pre-allocated 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	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.

Table 11. Sysplex data services return and reason codes (SMF services) (continued)			
Return code	**Reason code**	**Service routine**	**Meaning**
			Action
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.
12	37	Q,R,-,-	**Meaning:** Error - All SMF data buffers 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 data buffer is inactive.
			Action: Start SMF data buffers 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 z/OS Data Gatherer internal use.
			Action: Not applicable.
16	41	Q,-,-,-	**Meaning:** Severe error - The calling program specified an invalid value for the request type (*request_type*).
			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.

Table 11. Sysplex data services return and reason codes (SMF services) (continued)

Return code	Reason code	Service routine	Meaning
			Action
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,-,2,-	**Meaning:** Severe error - An invalid SMF record type or subtype was specified in the record type list (*smf_record_type_list*) for the ERBDSQRY or ERB2XDGS 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.
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 - The service 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 - The service 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.

Return code	Reason code	Service routine	Meaning
			Action
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 sysplex data services for SMF data (ERBDSQRY, ERBDSREC, ERB2XDGS and ERB3XDRS).
			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	84	-,-,2,-	**Meaning:** Severe error - The user is not authorized to access Monitor II data.
			Action: Contact your local security administrator.
16	85	-,-,-,3	**Meaning:** Severe error - The user is not authorized to access Monitor III data.
			Action: Contact your local security administrator.
16	86	-,-,2,-	**Meaning:** Severe error - The calling program is not authorized or is using a data reduction exit that is not approved.
			Action: If the calling program is properly designed to be safe to run authorized, the application or program can be adapted to run in supervisor state, system state, or APF authorized. Otherwise, approve the exit by adding RACF resource ERBSDS.MON2EXIT.*exit_name* to RACLISTed class FACILITY and grant read access to the profile for the user ID invoking the Sysplex Data Server API.
16	87	-,-,-,3	**Meaning:** Severe error - The calling program is not authorized or is using a data reduction exit that is not approved.
			Action: If the calling program is properly designed to be safe to run authorized, the application or program can be adapted to run in supervisor state, system state, or APF authorized. Otherwise, approve the exit by adding RACF resource ERBSDS.MON3EXIT.*exit_name* to RACLISTed class FACILITY and grant read access to the profile for the user ID invoking the Sysplex Data Server API.
16	90	Q,R,2,3	**Meaning:** Severe error - z/OS Data Gatherer encountered a severe error when calling the service routine. This may be caused by a terminating data gatherer address space.
			Action: Restart z/OS Data Gatherer and rerun your program.
16	91	-,-,2,3	**Meaning:** Severe error - z/OS Data Gatherer 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.

Table 11. Sysplex data services return and reason codes (SMF services) (continued)			
Return code	**Reason code**	**Service routine**	**Meaning**
			Action
16	92	-,-,2,3	**Meaning:** Severe error - z/OS Data Gatherer recognized a severe error when executing the service exit routine. The exit completion code is provided in the answer area returned by the service. This error can be caused by a disabled RMF feature on one system in a Parallel Sysplex. An example of such a misconfiguration is that a Monitor III Sysplex report is requested on one system that has the RMF feature enabled, but other systems in the Parallel Sysplex do not have the RMF feature enabled. See *z/OS Planning for Installation* for more information.
			Action: Correct the exit routine problems and rerun your program.
20	0	Q,R,2,3	**Meaning:** Unrecoverable error - An unrecoverable 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.

Layout of callable services answer area

When the sysplex data services 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:

where:

NAM
Four-character acronym of the common header as follows:

- 'DSQA' for ERBDSQRY/ERBDSQ64
- 'DSRA' for ERBDSREC/ERBDSR64
- 'XDGH' for ERB2XDGS/ERB2XD64
- 'XDRH' for ERB3XDRS/ERB3XD64

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 z/OS Data Gatherer is not active on this system, this field contains hex zeros.

    RMFn
        32-bit 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 data buffer for SMF data on this system ('1'B = active).
        - Bit 2 indicates the status of the Monitor III address space on this system ('1'B = active).
        - Bits 3 to 31 are reserved.

ERBDSQRY/ERBDSQ64 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:

RECTOK1	SMFHDR1...
SMFHDR1 (cont.)	
RECTOK2	SMFHDR2...
SMFHDR2 (cont.)	
...	
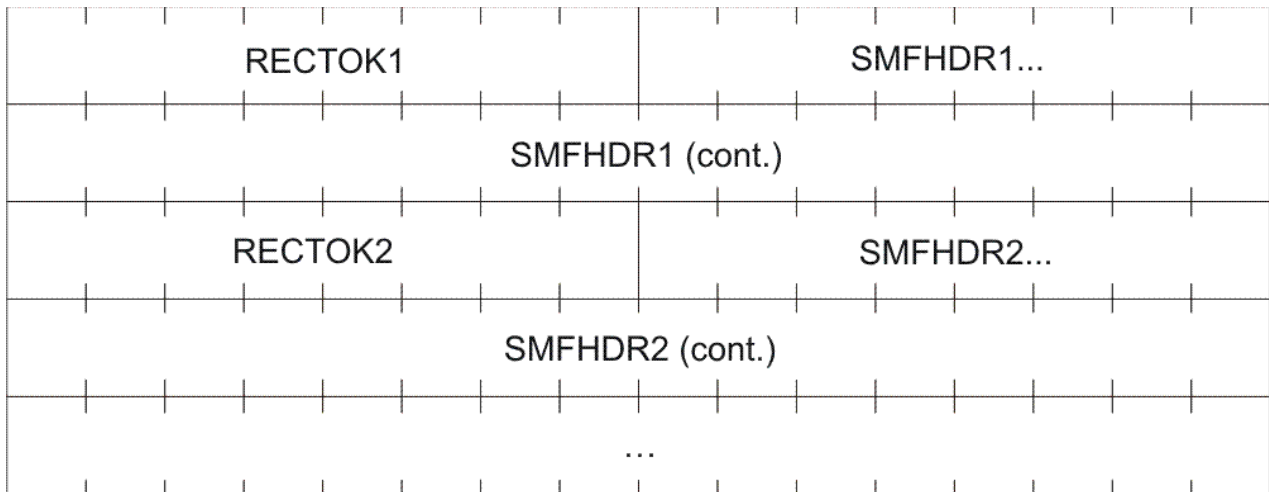
where:

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	0*CYYDDD*F	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:

RECTOK1	SMFHDR1...
SMFHDR1 (cont.)	
RMFINFO1	
RECTOK2	SMFHDR2...
SMFHDR2 (cont.)	
RMFINFO2	
...	
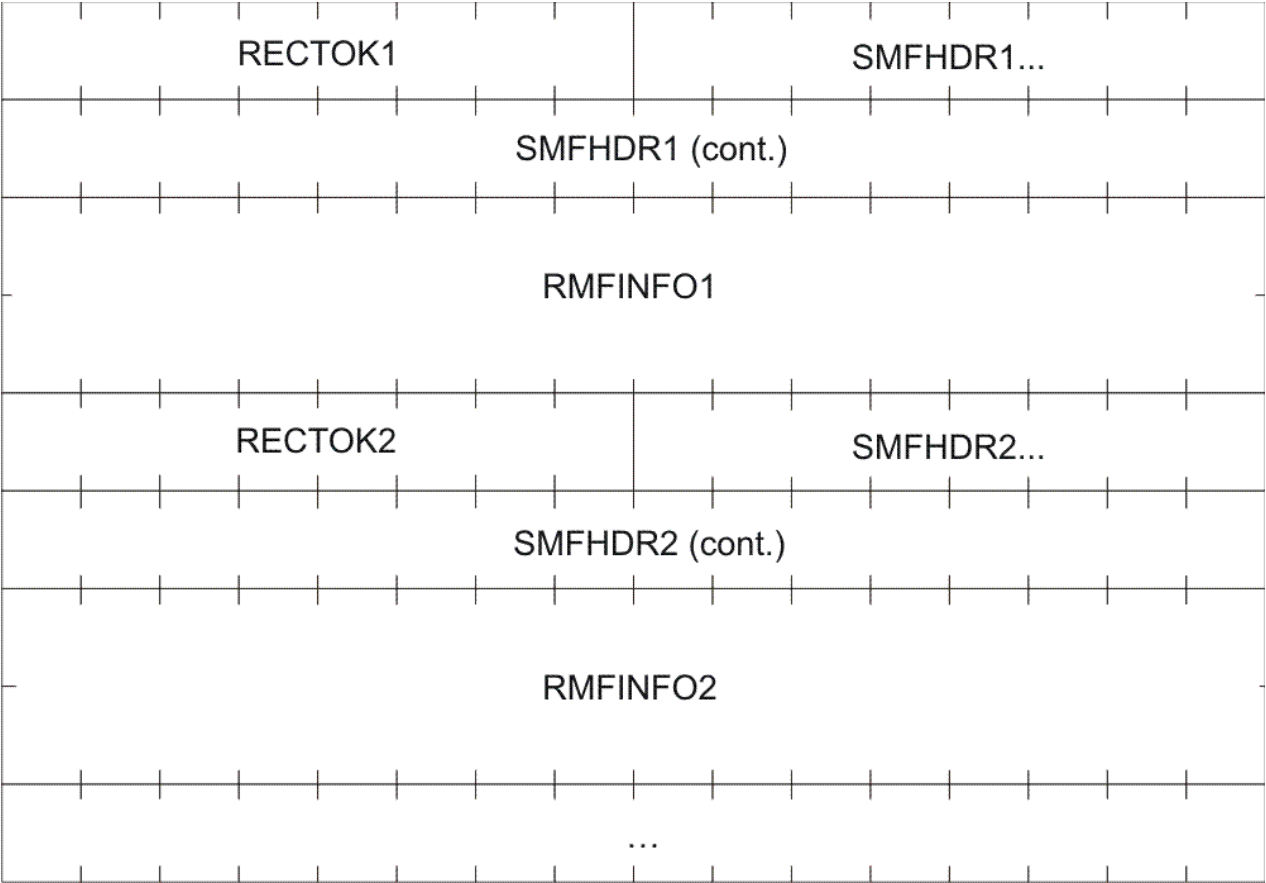
where:

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	Measurement interval start date
SMFxxIST	4	0HHMMSSF	Measurement interval start time
SMFxxINT	4	MMSSTTTF	Measurement interval length (in seconds)
SMFxxOIL	2	Integer	Projected measurement interval length (in seconds)
SMFxxSYN	2	Integer	Measurement interval synchronization value (in seconds)
SMFxxLGO	8	(STCK)	GMT offset to local time (in STCK format)
SMFxxGIE	8	(STCK)	Projected measurement interval end GMT time (in STCK format)

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

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

ERBDSREC/ERBDSR64 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.

The complete layout of the data section is as follows:

RL1	RH1	RC1	*
RECTOK1		SMFRECORD1.	

SMFRECORD1 (cont.)

RL2	RH2	RC2	*
RECTOK2		SMFRECORD2...	

SMFRECORD2 (cont.)

. . .
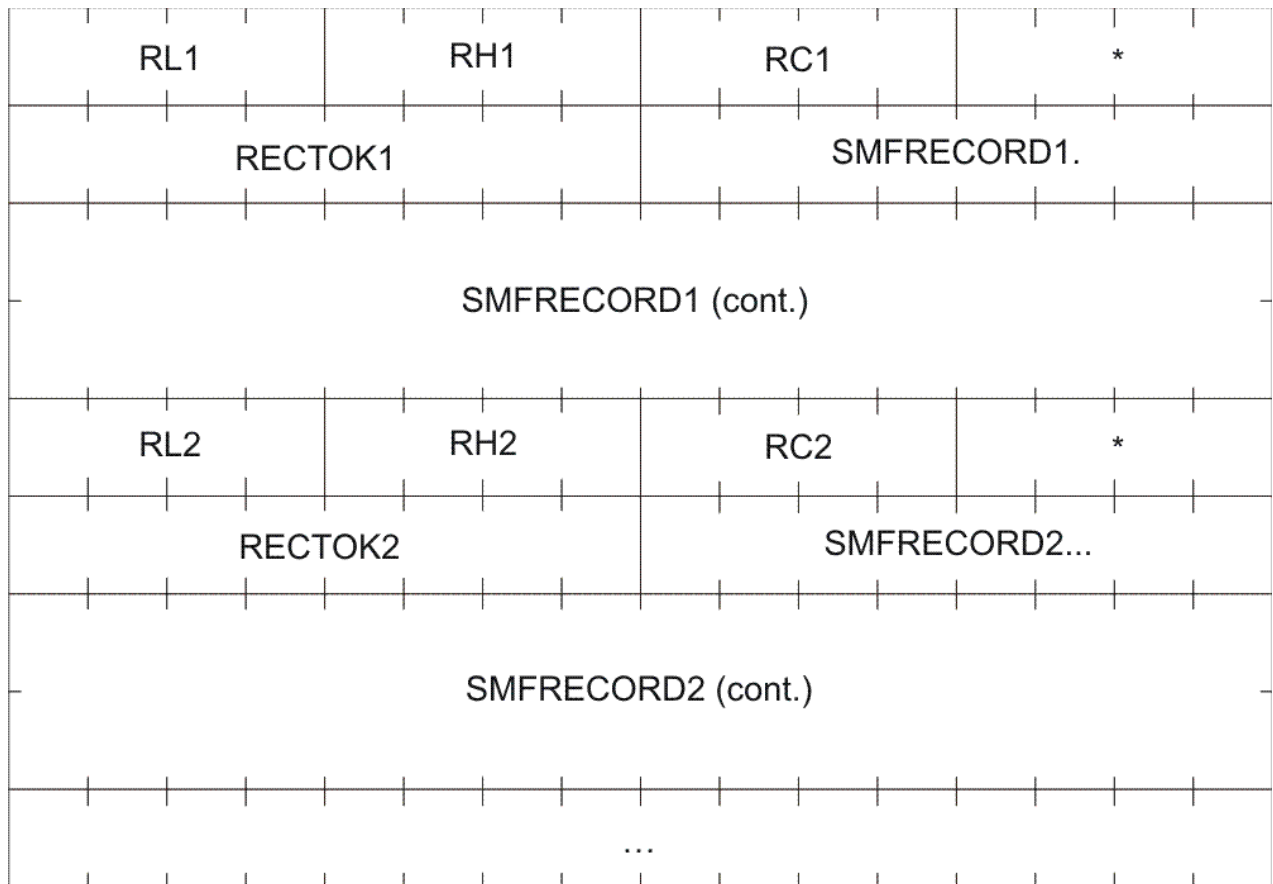
where:

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/ERB2XD64 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

DEL	HDL	RTN	RSN
CPU	PRT	DRC	DRC
…	SRM	SID	SRC
ZAP	ZIP		

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

SID
SMF system ID.

SRC
System return code.

If performance data for more than 65535 DASD devices (e.g. *dg_options = DASD)* is requested in a IBM zHyperWrite environment, only performance data for a maximum of 65535 devices are passed to the data reduction exit routine. This condition is indicated by return code 64 provided in this field.

ZAP
 System CPU utilization of zAAPs (if Monitor I CPU gathering is not active, this field has the value '-1')
ZIP
 System CPU utilization of zIIPs (if Monitor I CPU gathering is not active, this field has the value '-1')

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

ERB3XDRS/ERB3XD64 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 data header is

DEL	HDL	RTN	RSN
DGV	*	DGS	MNT
SAM	RNG	BEG	
…	END		
…	DRC		
DSG		DEG	
DIT		DFA	
DLA		…	

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 Monitor III 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.

Chapter 3. Accessing data using the z/OS Data Gatherer REST services

The information in the following topics enable you to work with the z/OS Data Gatherer REST services.

SMF REST services

SMF REST endpoints for the following structures are supported:

- SMF record types 70–79
- SMF record type 99, subtypes 1, 2, 6, 12, and 14
- SMF record type 113

Accessing the SMF REST services OpenAPI document

The OpenAPI document for the SMF REST services can be accessed at https://*host*:*port*/zosmf/zosdg/smf/v3/api-docs. The OpenAPI document provides in-place documentation about all the possible HTTP requests, including request paths, associated query parameters, and possible responses.

How to specify requests to the SMF REST services

Callers interact with the SMF REST services via HTTP REST calls. Possible callers are command-line based (such as **curl**), a web browser, front-end applications, or programmatic callers created via OpenAPI Generator. The base URI for the SMF REST services is https://*host*:*port*/zosmf/zosdg/smf/*version*.

For version v1, SMF REST services calls can be separated into 2 categories. For both v1 call categories, all available request paths and responses can be found in the OpenAPI document.

- The first category is a *Discover SMF Record Data* call, which uses the https://*host*:*port*/zosmf/zosdg/smf/v1/smf/discover URI.

  Example: The following URL is an example of a discover call using the HTTP GET method:

  ```
  https://host:port/zosmf/zosdg/smf/v1/smf/discover/dataset/exists/datasetName
  ```

  The response is a JSON document that contains either true or false, depending on whether the data set (*datasetName*) exists.

- The second category is a *Read SMF Record Data* call, which uses the https://*host*:*port*/zosmf/zosdg/smf/v1/smf/type URI. The response is a JSON document with an array of objects. Each object represents an SMF record of a specific type and subtype. Each object includes all structures (record sections) and fields that were requested and available.

  Example 1: The following URL is an example of a read call for SMF type 70 subtype 1 with a minimal set of parameters using the HTTP GET method:

  ```
  https://host:port/zosmf/zosdg/smf/v1/smf/type/70/subtype/1?datasetName=SMF.DATASET.NAME
  ```

  The response is a JSON document that contains an array of objects that represent all SMF type 70 subtype 1 records, unfiltered, from the SMF.DATASET.NAME data set.

  Example 2: The following example shows a read call with more extensive use of additional optional parameters to filter the data, with each parameter shown on a new line for better visibility:

  ```
  https://host:port/zosmf/zosdg/smf/v1/smf/type/70/subtype/1
  ?datasetName=SMF.DATASET.NAME
  &systemName=ESYS
  &startTime=2022-03-28T10:00
  &endTime=2022-03-28T11:10:30
  &SMF70_SUBTYPE1=SMF70TME
  ```

```
&SMF70_SUBTYPE1_CPU_DATA=SMF70WAT
&SMF70_SUBTYPE1_CPU_DATA=SMF70CID
&SMF70_SUBTYPE1_CPU_IDENTIFICATION=*
```

The response is a JSON document that contains an array of objects that represent selected SMF type 70 subtype 1 records. From the SMF.DATASET.NAME data set, only records written by the ESYS system are of interest. With the additional **startTime** and **endTime** parameters, the data set is filtered for records in the 1-hour interval between 2022-03-28T10:00 and 2022-03-28T11:10:30. From the SMF70_SUBTYPE1 structure (effectively, the SMF type 70 subtype 1 header section), only the SMF70TME field is requested. For the SMF70_SUBTYPE1_CPU_DATA structure, the SMF70WAT and SMF70CID fields are requested. For the SMF70_SUBTYPE1_CPU_IDENTIFICATION structure, all fields (*) are requested.

In the JSON response, z/OS-specific data types used in SMF records are translated to common data types, as listed in Table 12 on page 50.

Table 12. Translation of z/OS-specific data types to common data types — Selected data types

z/OS data type	Description	Common type	Output examples
10010010	Usually flags	Mostly boolean	true false
[0-9]+D?	Packed numeric	Numeric	0 2 300
0YYYYDDD	Date format	ISO-date	2022-03-28
0CYYDDDF	Date format, where C indicates centuries since 1900 • C = 0 for the range 1900–1999 • C = 1 for the range 2000–2099	ISO-date	2022-03-28
TOD clock	Date and time format	ISO-date and time ISO-date ISO-time Integer (such as for durations)	2022-03-28T10:00:00.001 2022-03-28 10:00:00
HHMMSSth 0HHMMSSF MMSSTTTF 000TTTTF	Time formats	ISO-time	10:00:00.001 00:10:12.456 00:00:00.123 10:00:00 10:00

The following figures show example outputs using the common data types.

In Figure 8 on page 51, SMF records are modeled as indexed JSON objects in the output. z/OS-specific data types are translated to common data types. One-to-one relationships between SMF header and substructures, such as the Product Section, are also visible here.

```
▶ 0:                              {…}          SMF70TNS:                      0
▶ 1:                              {…}          SMF70TNL:                      0
▼ 2:                                           SMF70TNN:                      0
      SMF70LEN:                   1520       ▶ smf70Subtype1ProductSection:   {…}
      SMF70SEG:                   0          ▶ smf70Subtype1AsidArea:         {…}
      SMF70FLG:                   "11011110" ▼ smf70Subtype1CpuControl:
      SMF70RRF:                   true            SMF70MOD:                   "2964"
      SMF70SUT:                   true            SMF70VER:                   255
      SMF70V4:                    true            SMF70BNP:                   0
      SMF70ESA:                   true            SMF70INB:                   "00000000"
      SMF70VXA:                   true            SMF70DIF:                   false
      SMF70OS:                    true            SMF70NPC:                   false
      SMF70BFY:                   false           SMF70TSC:                   false
      SMF70RTY:                   70              SMF70PHY:                   false
      SMF70TME:                   "14:59:00.01"   SMF70DGE:                   false
      SMF70DTE:                   "2017-06-21"    SMF70VMG:                   false
      SMF70SID:                                   SMF70STF:                   "10011000"
      SMF70SSI:                   "RMF "          SMF70STS:                   true
      SMF70STY:                   1               SMF70ADC:                   false
      SMF70TRN:                   9               SMF70WUC:                   false
      SMF70PRS:                   100             SMF70RCU:                   true
      SMF70PRL:                   104             SMF70HWV:                   true
      SMF70PRN:                   1               SMF70PTC:                   false
      SMF70CCS:                   204             SMF70PLC:                   false
      SMF70CCL:                   344             SMF70GAV:                   false
      SMF70CCN:                   1               SMF70GTS:                   0
      SMF70CPS:                   548             SMF70MDL:                   "757         "
      SMF70CPL:                   92              SMF70DSA:                   12
      SMF70CPN:                   2               SMF70IFA:                   0
      SMF70ASS:                   732             SMF70CPA:                   506
      SMF70ASL:                   788             SMF70WLA:                   1024
      SMF70ASN:                   1               SMF70LAC:                   2
      SMF70BCS:                   1520            SMF70HWM:                   "N63         "
      SMF70BCL:                   84              SMF70SUP:                   0
      SMF70BCN:                   0               SMF70GJT:                   "02:00:00"
      SMF70BVS:                   1520            SMF70POM:                   "02  "
```

Figure 8. Example JSON documents in response to Read SMF Record Data calls (1-to-1 relationships)

shows the responses from 2 Read SMF Record Data calls. The response to the first call (the first 2 columns) shows 1-to-many relationships for the SMF type 70 subtype 1 CPU Data Section. The response to the second call (the last 2 columns) shows formatted fields of SMF type 78 subtype 2 as expanded substructures.

Figure 9. Example JSON documents in response to Read SMF Record Data calls (1-to-many-relationships, formatted fields)

Monitor II REST services

Monitor II REST endpoints for IBM Function Registry for z/OS data are supported from one system or all systems in a sysplex.

Accessing the Monitor II REST services OpenAPI document

The OpenAPI document for the SMF REST services can be accessed at https://*host*:*port*/zosmf/ zosdg/m2/v3/api-docs. The OpenAPI document provides in-place documentation about all the possible HTTP requests, including request paths, associated query parameters, and possible responses.

How to specify requests to the Monitor II REST services

Callers interact with the Monitor II REST services via HTTP REST calls. Possible callers are command-line based (such as **curl**), a web browser, front-end applications, or programmatic callers created

via OpenAPI Generator. The base URI for the SMF REST services is https://*host*:*port*/zosmf/zosdg/m2/*version*/.

All available request paths and responses can be found in the OpenAPI document.

Chapter 4. Adding Monitor I installation exits

The following topics describe how to create and add Monitor I user exit routines.

Overview

Facilities in z/OS Data Gatherer and 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 Monitor I cycle, collect this data and examine system indicators at each Monitor I interval, format and write your own SMF records, and format and write your own reports.

Note: The topics that follow only describe the z/OS Data Gatherer aspects.

Monitor I session gatherer user exits

To gather data relevant to your installation during a Monitor I session, z/OS Data Gatherer provides both the EXITS option and installation exits at various points during Monitor I session processing. When EXITS is specified, you can perform the following functions:

- Initialize for the other use exit routines.
- Sample fixed CSA, SQA, or nucleus data at each Monitor I cycle.
- Perform interval processing, such as reduce sampled data, examine system state indicators, and format SMF records to be written to the SMF data set or passed to your report writer.
- Handle termination processing for the other installation exits.

Guidelines for Monitor I user exit routines

The following guidelines apply to Monitor I user exit routines:

- All user exit routines must be reentrant.
- 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 Monitor I passes to each user exit routine is the address of a 2-word area that is reserved for the use of your routines. Because this area provides a means of communication between your exit routines, its use should be governed by conventions that are agreed upon by your installation.
- Monitor I passes a phase parameter to each user exit routine except the sampler user exit. This phase parameter indicates which data gatherer phase is invoking the user exit.

z/OS Data Gatherer provides dummy routines for all Monitor I session exits that are not used.

⚠️ **CAUTION:** Because all of the user exit routines 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 cause a system outage.

The user functions that your exit routines can perform are described in the topics that follow.

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 that the other installation exits require, such as building a control block structure.

When the exit routine gets control, register 1 points to a 3-word address list. The first address points to the 2-word area reserved for use by your routines. This same 2-word area is passed to all the user

exit routines and can be used for communication between them. The second address points to the data gatherer phase parameter, a fullword 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 10 on page 56 illustrates the input parameter structure.
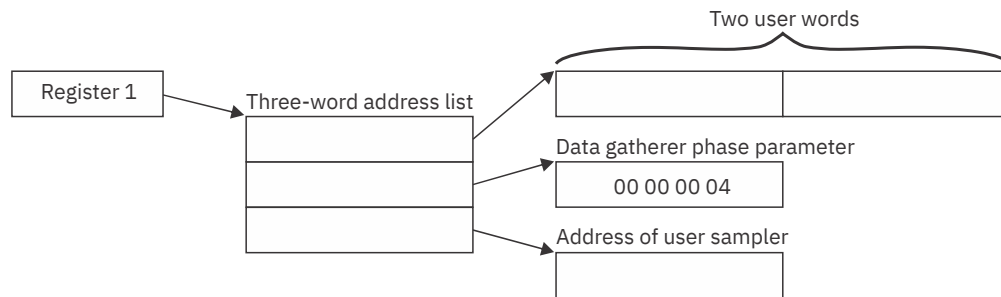
Figure 10. ERBMFIUC input parameter structure

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" on page 56. When you have a user sampler, your initialization routine must satisfy the following requirements:

- 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 that the sampler routine will require must be obtained; this storage must be obtained from SQA (subpool 245).
- The address of the obtained SQA storage must be placed in one of the 2 user words. The choice depends on the conventions established at your installation.

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

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 and 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 Monitor I to invoke your user sampler.

When the sampler gets control, register 1 points to a 2-word area. One of these words, selected by your installation, contains the address of the storage area obtained for the sampler by ERBMFIUC. Figure 11 on page 56 illustrates the input parameter structure.
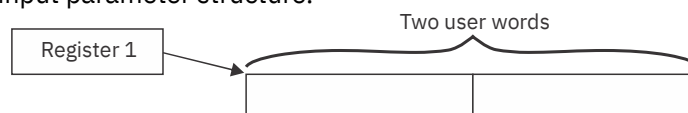
Figure 11. User sampler input parameter structure

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 Monitor I interval, z/OS Data Gatherer 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 system completion (abend) code of 0FE.

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

Note: The user sampler must reside in SYS1.SGRBLPA. For more information, see "Adding your data gatherer routines to Monitor I" on page 58.

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 Monitor I interval.

When the exit gets control, register 1 points to a 2-word address list. The first address points to the 2-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 data gatherer 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 a Monitor I interval, or X'C' when the exit is called at the end of a Monitor I interval for which data collection was skipped. Figure 12 on page 57 illustrates the input parameter structure.
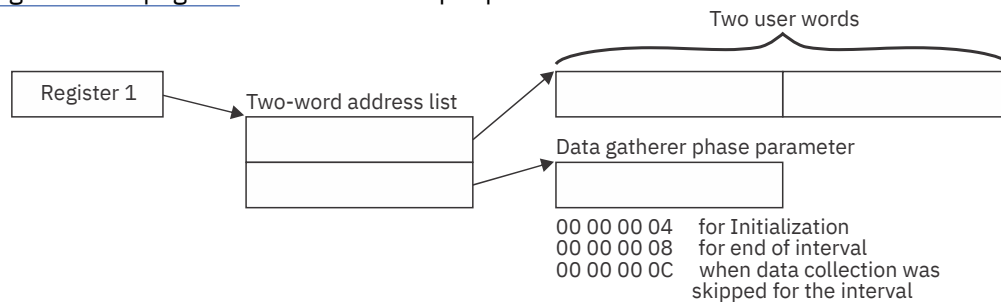
Figure 12. ERBMFDUC input parameter structure

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 SMFEWTM - Writing SMF records in *z/OS MVS System Management Facilities (SMF)*.

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

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 2-word address list. The first address points to a 2-word area reserved for use by your routines. The second address points to the data gatherer phase parameter, which is always X'C' for termination. Figure 13 on page 57 illustrates the input parameter structure.
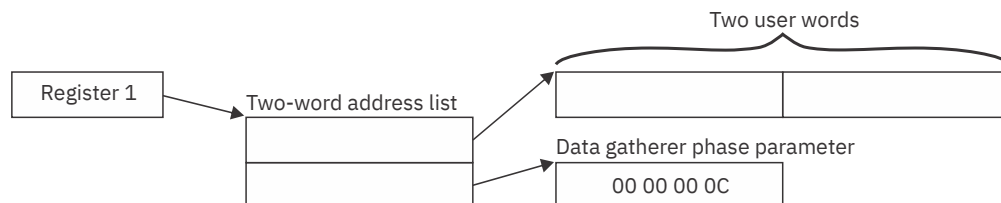
Figure 13. ERBMFTUR input parameter structure

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 samples or data areas and to free any user SQA data areas obtained by the other exits.

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

Adding your data gatherer routines to Monitor I

Before your Monitor I session user exit routines can be tested and used, they must be assembled and link edited with the appropriate z/OS Data Gatherer modules. If you are using your private libraries, ensure that they are concatenated ahead of the distributed z/OS Data Gatherer libraries. shows sample JCL for performing the required link edit for all user routines except the sampler routine.

```
//LINKEXIT JOB MSGLEVEL=1
//LINK0001 EXEC PGM=IEWL,PARM='MAP,XREF,REUS,RENT,REFR,NCAL'
//SYSPRINT DD SYSOUT=A
//SYSLMOD DD DSN=SYS1.SGRBLINK,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)
    (ERBMFTUR object deck)
    ENTRY ERBMFTUR
    NAME ERBMFTUR(R)
 /*
```

Figure 14. Replacing installation exits

If you have a user sampler, a separate link edit is required; a sample is shown in .

```
//LINKEXIT JOB MSGLEVEL=1
//LINK0001 EXEC PGM=IEWL,PARM='MAP,XREF,REUS,RENT,REFR,NCAL'
//SYSPRINT DD SYSOUT=A
//SYSLMOD DD DSN=SYS1.SGRBLPA,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 15. Adding a user sampler

Chapter 5. Using Monitor III VSAM data set support

This topic provides the following information:

- It describes the data set structure and content for the Monitor III data set support function
- It lists the record fields and table entries associated with data set support

See *z/OS Data Gatherer User's Guide* for more information about data set support and recording.

Data set record structure

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, Monitor III uses VSAM relative record data sets (RRDS) to record measurement information during a Monitor III gatherer session.

During data set recording, Monitor III 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, Monitor III 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 decompression interface (ERB3RDEC) is used to decompress the data one MINTIME at a time. The 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 for more information.

The Monitor III 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 the data gatherer 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). The Monitor III data gatherer builds one index entry for each MINTIME set of samples in the data set. When no more entries can fit into the index, the data gatherer closes the data set. The records in the data set following the index information contain the measurements of each MINTIME set of samples (see). Monitor III 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

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 16. 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 Monitor III 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.

Data set decompression

The MINTIME set-of-samples stored on VSAM data sets is compressed by Monitor III 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 . 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 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 <u>Table 13 on page 61</u>.

Table 13. 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
```

Data set content

A MINTIME set of samples collected during the Monitor III gatherer session is independent of other MINTIME sets of samples. Measurement values for each MINTIME set of samples are organized as tables or records, the formats of which appear in "Monitor III data set record and table formats" on page 66.

The types of measurement tables or records are:

ERBASIG3
    ASID table

ERBCATG3
    Cache data information table

ERBCFIG3
    Coupling facility information table

ERBCPCDB
    CPC data control block

ERBCPDG3
    Channel data information table

ERBCPUDB
    CPU data block

ERBCPUG3
Processor data control block

ERBCRYG3
Cryptographic hardware data table

ERBCSRG3
Common storage remaining table

ERBDDNG3
Device data set name list

ERBDSIG3
Data set header and index

ERBDVTG3
Device table

ERBENCG3
Enclave data table

ERBENTG3
Enqueue name table

ERBGEIG3
General information table

ERBIQDG3
I/O queuing performance data table

ERBLOKG3
Lock performance data table

ERBOPDG3
OMVS process data table

ERBPCIG3
PCIE activity data table

ERBRCDG3
Resource collection data table

ERBREDG3
Resource data record

ERBSCMG3
Extended Asynchronous Data Mover (EADM) data table (includes storage class memory (SCM) data)

ERBSHDG3
Sample header

ERBSPGG3
Storage group and volume data table

ERBSSHG3
MINTIME set of samples header

ERBSVPG3
Service policy data table

ERBUWDG3
USE/WAIT record

ERBVRIG3
VSAM RLS information data table

ERBXCFG3
XCF activity data table

ERBXMHG3
Moved samples header control block

ERBZFXG3
zFS performance data table

Each is described in "Monitor III data set record and table formats" on page 66. 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 17 on page 64 shows the relationships between the Monitor III data set support tables and records.

Figure 17. Monitor III Measurement Table and Record Relationships
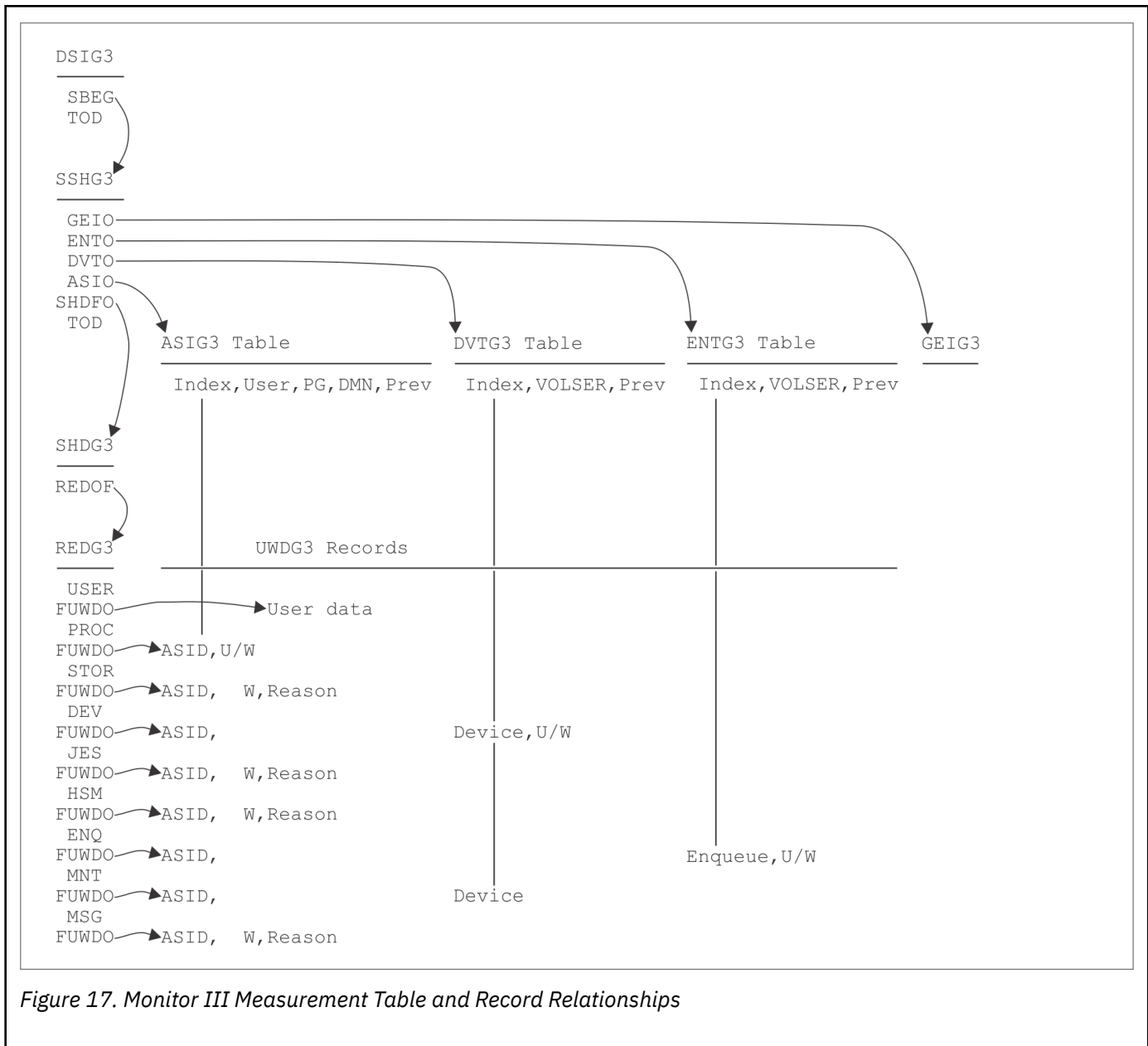
The data set header and index (ERBDSIG3) describe the available measurement times (MINTIME sets of samples) and the data set offsets of each MINTIME set of samples header (ERBSSHG3).

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), and the common storage remaining table (ERBCSRG3). 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. The Monitor III data gatherer first samples each type of hardware and software resource; the Monitor III data gatherer

then samples user-written exit routines. The sample header (ERBSHDG3) for user-written exit routines contains an offset to the first resource record.

The Monitor III data gatherer 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 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.

The Monitor III data gatherer 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 17 on page 64).

```
Index × 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 17 on page 64) minus 1:

```
(Index – 1) × 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.

Monitor III data set record and table formats

This topic describes the measurement records and tables used for the Monitor III data set support function. Fields that are reserved for Monitor III data gatherer are used for debugging purposes, for maintaining the data areas, or do not contain data for RMF Monitor III reports.

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

Dec offset	Hex offset	Name	Length	Format	Description
ASIG3 header section:					
0	0	ASIASIG3	5	EBCDIC	Acronym 'ASIG3'
5	5	ASIVERG3	1	binary	ASIG3 version X'1E'
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	ASIFLAG	1	binary	ASIG3 flags **Bit** **Meaning when set** **0** ASIG3 converted to lower service level **1** ASIG3 converted to higher release or service level **2–7** Reserved
25	19	*	7	*	Reserved
32	20	ASIENTRY(*)	*	*	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	Job name for this address space ID (ASID). This and the next 5 offsets describe the sort criteria for the address space (ASID). Monitor III creates a new entry whenever the JOBNAME 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

Dec offset	Hex offset	Name	Length	Format	Description
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	Reserved
48	30	ASIESFI_VE	4	floating point	Reserved
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

Dec offset	Hex offset	Name	Length	Format	Description
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	Number of single state samples delayed by XCF - part of subs
120	78	ASIJCLAS	8	EBCDIC	Job class, Source: OUCBCLS
128	80	ASIPINES	4	binary	Reserved
132	84	ASIFLAG2	4	binary	Common storage flags **Bit** **Meaning When Set** **0** CSA and RUCSA 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

Dec offset	Hex offset	Name	Length	Format	Description
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
188	BC	ASIIOCNT	4	binary	IO count
192	C0	ASILSCT	2	binary	Count of "long" logical swaps
194	C2	ASIESCT	2	binary	Reserved
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 2: Terminal input wait
216	D8	ASIORTO	2	binary	STOR/OUTR delay samples for swap reason 1: 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. No longer used - refer to ASIORMP.
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

Dec offset	Hex offset	Name	Length	Format	Description
244	F4	ASIORIW	2	binary	STOR/OUTR delay samples for swap reason 16: OMVS input
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	Number of 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

Dec offset	Hex offset	Name	Length	Format	Description
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** Address space is a CICS TOR that matched a classification rule in the active policy which allows managing the region based on the region goals but also ensures that completed transactions are reported and used for management of the CICS AORs **7** I/O priority group HIGH was assigned either to the address space or to transaction service classes served by the address space **8** Address space is currently associated with a tenant resource group. **9** Reserved. **10** Address space matched a classification rule in the active policy which enables for recovery process boost. **11–31** Reserved.
328	148	ASISUCIF	2	binary	Number of single state samples using zAAP
330	14A	ASISUCIC	2	binary	Number of single state samples using zAAP on CP
332	14C	ASISDCIF	2	binary	Number of single state samples delayed by zAAP
334	14E	ASISDCCP	2	binary	Number of single state samples delayed by standard CP
336	150	ASICPTA	4	binary	Accumulated CPU time
340	154	ASIIFATA	4	binary	Accumulated zAAP time
344	158	ASIIFCTA	4	binary	Accumulated zAAP on CP time
348	15C	ASIMCUSE	4	binary	Multi state processor using count

ERBASIG3 ASID table

Dec offset	Hex offset	Name	Length	Format	Description
352	160	ASIMCDLY	4	binary	Multi state processor delay count
356	164	ASISUCCP	2	binary	Number of single state samples using standard CP
358	166	ASISUCSP	2	binary	Number of single state samples using zIIP
360	168	ASISUCSC	2	binary	Number of single state samples using zIIP on CP
362	16A	ASISDCSP	2	binary	Number of single state samples delayed by zIIP
364	16C	ASI_TIME_ON_ZIIP	4	binary	Accumulated zIIP time
368	170	ASI_ZIIP_TIME_ON_CP	4	binary	Accumulated zIIP on CP time
372	174	ASI_IFA_PHTM	4	binary	zAAP-only equivalent of ASIPHTMA
376	178	ASI_ZIIP_PHTM	4	binary	zIIP-only equivalent of ASIPHTMA
380	17C	ASI_LargeMemoryObjects	4	floating point	Number of fixed 1 MB memory objects allocated [1]
384	180	ASI_LargePagesBackedInReal	4	floating point	Number of 1 MB pages fixed in central storage [1]
388	184	ASI_LVNMOMB	4	floating point	Number of high virtual private memory objects allocated [1]
392	188	ASI_HVCommonNMOMB	4	floating point	Number of high virtual common memory objects allocated [1]
396	18B	ASI_LVSHRNMOMB	4	floating point	Number of high virtual shared memory objects allocated [1]
400	190	ASI_LVABytes	8	floating point	Amount of storage allocated from high virtual private memory in memory objects [1]
408	198	ASI_HVCommonBytes	8	floating point	Amount of storage allocated from high virtual common memory in memory objects [1]
416	1A0	ASI_LVSHRBytes	8	floating point	Amount of storage in shared memory objects owned by this address space.
424	1A8	ASI_HVCommonHWMBytes	8	floating point	High water mark for the amount of high virtual common storage allocated [1]
432	1B0	ASI_LVMemLim	8	floating point	Address space memory limit in MB [1]
440	1B8	*	376	*	Reserved
816	330	ASIQScanReq	8	binary	Number of QScan requests issued by this address space
824	338	ASIQScanSpecReq	8	binary	Number of specific QScan requests issued by this address space
832	340	ASIQScanRes	8	binary	Number of resources returned by QScan requests
840	348	ASIQScanResSq	16	binary	Sum of the squares of ASIQScanRes
856	358	ASIQScanTime	8	binary	Sum of QScan request times in microseconds
864	360	ASIQScanTimeSq	16	binary	Sum of the squares of ASIQScanTime
880	370	ASI_1MBFixedFrames	4	floating point	Number of 1 MB page-fixed frames used for pageable/DREF memory objects [1]
884	374	ASI_1MBPageableFrames	4	floating point	Number of 1 MB pageable/DREF frames[1]
888	378	*	112	*	Reserved
1000	3E8	ASIDP	2	binary	Dispatching priority
1002	3EA	*	2	*	Reserved
1004	3EC	ASITRT	4	binary	Transaction resident time

Dec offset	Hex offset	Name	Length	Format	Description
1008	3F0	ASITRCA_S	4	binary	Total accumulated number of transactions (sum)
1012	3F4	ASIDCTIA_S	4	binary	Total accumulated channel connect time (sum)
1016	3F8	ASIIOCNT_S	4	binary	Total EXCP count (sum)
1020	3FC	*	4	*	Reserved
1024	400	ASICPUTA_LF	8	floating point	Total accumulated CPU time in milliseconds
1032	408	ASITCBTA_LF	8	floating point	Total accumulated TCB time in milliseconds
1040	410	ASIFRXH_LF	8	floating point	Number of fixed frames high[1]
1048	418	ASIFRXA_LF	8	floating point	Number of fixed frames above[1]
1056	420	ASIFRXB_LF	8	floating point	Number of fixed frames below[1]
1064	428	ASI_HvShrPageValidations	4	floating point	Number of page validations for high virtual shared
1068	42C	ASI_LvShr1MNMomb	8	floating point	Number of shared 1M memory objects allocated
1076	434	ASI_LvShr4KB	8	floating point	Number of shared bytes from high virtual memory in units of 4K
1084	43C	ASI_LvShr1MGBytes	8	floating point	High water mark for the amount of storage allocated in shared memory objects.
1092	444	ASI_FreemainedFrames	8	floating point	Number of FREEMAINed frames
1100	44C	*	4	*	Reserved
1104	450	ASISTAFL	4	binary	Status Flags
1108	454	*	4	*	Reserved
1112	458	ASI_2GMemoryObjects	8	floating point	Number of fixed 2 GB memory objects allocated[1]
1120	460	ASI_2GPagesBackedInReal	8	floating point	Number of 2 GB pages fixed in central storage [1]
1128	468	ASIORMP	2	binary	STOR/OUTR delay samples for swap reason 7: Memory pool shortage
1130	46A	*	6	binary	Reserved
1136	470	ASIEXTNA	36	binary	Reserved
1172	494	ASI_EJST	4	binary	TCB processor time (all processor types)
1176	498	*	4	*	Reserved
1180	49C	ASI_SRBT	4	binary	Non-preemptable SRB time (all processor types)
1184	4A0	*	12	*	Reserved
1196	4AC	ASICPUTA_CP	4	binary	All non-enclave time on CP, which includes TCB, non-preemptable SRB, client SRB, other SRB
1200	4B0	*	12	*	Reserved
1212	4BC	ASI_CP_PHTM	4	binary	Preemptable-class SRB time on CP, which includes time for all types of preemptable SRBs (PSRB, CRSRB, ESRB, ETCB)
1216	4C0	*	16	*	Reserved

Dec offset	Hex offset	Name	Length	Format	Description
1232	4D0	ASIRUCSAA	4	floating point	RUCSA allocation
1236	4D4	ASIERUCSAA	4	floating point	ERUCSA allocation
1240	4D8	*	8	*	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

Dec offset	Hex offset	Name	Length	Format	Description
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	*	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	*	Reserved
16	10	CATG3_Hdr_Len	2	binary	Length of CATG3 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	*	Reserved
48	30	CATG3_LDTO	8	binary	Offset GMT to local time (STCK format)
56	38	*	8	*	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	*	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	*	Reserved
29	1D	CATG3_CMPC	3	binary	Bit 0-11: System completion code Bit 12-23: User completion code

Dec offset	Hex offset	Name	Length	Format	Description
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

Dec offset	Hex offset	Name	Length	Format	Description
CFIG3 header section:					
0	0	CFICFIG3	5	EBCDIC	Acronym 'CFIG3'
5	5	CFIVERG3	1	binary	CFIG3 version X'09'
6	6	CFIHDRLE	2	binary	Length of CFIG3 header
8	8	CFITOTLE	4	binary	Length 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	CFIRANGE	4	binary	Reporting range in seconds
44	2C	CFIPONAM	8	EBCDIC	Policy name
52	34	CFIPOACT	8	EBCDIC	Policy activation time
60	3C	CFIPREQS	4	binary	Policy is not large enough to contain all structures that exist in the CF
64	40	CFIPREQC	2	binary	Policy is not large enough to contain all connections
66	42	CFIFLAG	1	binary	CFIG3 flags **Bit** **Meaning when set** **0** CFIG3 converted to lower service level **1** CFIG3 converted to higher release or service level **2 - 7** Reserved
67	43	*	1	*	Reserved
68	44	CFICHPOF	4	binary	Offset to first channel path entry CFICHPAS

ERBCFIG3 - Coupling facility data

Dec offset	Hex offset	Name	Length	Format	Description
72	48	CFICHPLE	2	binary	Length of channel path entry
74	4A	CFICHPNR	2	binary	Number of channel path entries
76	4C	CFISCMOF	4	binary	Offset to first storage class memory entry CFISSCMS
80	50	CFISCMLE	2	binary	Length of storage class memory entry
82	52	CFISCMNR	2	binary	Number of storage class memory entries
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

Dec offset	Hex offset	Name	Length	Format	Description
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 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
					The CF storage is volatile when this bit = 1.
					6
					Policy change pending which will delete this coupling facility from the CFRM active policy when all allocated structures are gone from this coupling facility.
					7
					The coupling facility to CFRM policy reconcile process is in progress.
					8
					The coupling facility has failed.
					9
					CF dynamic dispatching.
					10
					Recovery manager active.
					11
					Maintenance mode active.
					12
					Coupling thin interrupts are enabled.
					13
					Optimized coupling facility hardware data gathering active.
					14 - 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

Dec offset	Hex offset	Name	Length	Format	Description
84	54	CFIENTID(8)	1	binary	CF link type. Valid entries have corresponding bit in CFIENPAM set
92	5C	CFIENSCN	4	binary	Connected MVS system counts - Number of systems connected to specified CF
96	60	CFIENSTI	4	binary	Structure count in policy - Number of records for structures in specified CF
100	64	CFIENSTO	4	binary	Structure count out policy - Number of structures in this CF which cannot be added to the policy
104	68	CFIENTCS	4	binary	Total control space in 4K blocks. Control space + non-control space = total space
108	6C	CFIENFCS	4	binary	Total free control space in 4K blocks
112	70	CFIENTDS	4	binary	Total dump space in 4K blocks
116	74	CFIENFDS	4	binary	Free dump space in 4K blocks
120	78	CFIENSCT	8	binary	Summed contention time in microseconds for waiting for subchannels to become free for synchronous immediate operations
128	80	CFIENFOC	4	binary	Count of the number of summed times - for unsuccessful operations
132	84	CFIENFOT	8	binary	Summed service time in microseconds of unsuccessful operations
140	8C	CFIENPDE	2	binary	Number of dedicated processors
142	8E	CFIENPSH	2	binary	Number of shared processors
144	90	CFIENPWG	2	binary	Shared processor average weight
146	92	CFIENPCO	1	binary	Path composite mask for CFIENPID
147	93	*	1	*	Reserved
148	94	CFIENCP1	4	binary	Index of first channel path entry CFICHPAS belonging to this CF
152	98	CFIENCP#	4	binary	Number of CFICHPAS entries belonging to this CF
156	9C	CFIENSC1	4	binary	Index of first SCM entry CFISSCM belonging to this CF
160	A0	CFIENSC#	4	binary	Number of CFISSCM entries belonging to this CF
164	A4	*	4	*	Reserved
168	A8	CFIENTSC	8	binary	Total CF storage class memory in 4K blocks
176	B0	CFIENFSC	8	binary	Free CF storage class memory in 4K blocks
184	B8	CFIENISM	2	binary	Storage class memory increment in 4K blocks
CFISTRUS table entry section:					
0	0	CFISTNAM	16	EBCDIC	Name of connected structure in this CF
16	10	CFISTVER	8	binary	Structure Version number
24	18	CFISTTYP	1	binary	Structure type: 1 = unserialized list 2 = serialized list 3 = lock 4 = cache 5 = unknown
25	19	*	3	*	Reserved

Dec offset	Hex offset	Name	Length	Format	Description

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
					    Async duplexing primary instance of structure
					3
					    Async duplexing secondary instance of structure
					4
					    Structure is encrypted
					5-7
					    Reserved

29	1D	*	1	*	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)
80	50	CFISTCOM	2	binary	Maximum number of connections allowed when structure was allocated in CF
82	52	*	2	*	Reserved
84	54	CFISTCOT	4	binary	Total of connections to the specified structure
88	58	CFISTCOP	4	binary	Number of connections to the specified structure with problems
92	5C	CFISTQTM	8	binary	Summed service time for queued requests in microseconds

Dec offset	Hex offset	Name	Length	Format	Description
100	64	CFISTFLE	2	binary	Status flags extended. **Bit** **Meaning When Set** **0** Only one structure allocated with this name **1** Rebuild (old): The original active structure is now the old structure **2** Rebuild (new): This structure is the new structure **3** Transitional state **4** Hold state **5** Structure cannot be deallocated since a dump table is associated with it **6** Structure failure for this version of the structure **7** Structure allocated with STRDISP = KEEP **8** Change pending in structure policy **9** Structure is defined in policy **10** Structure contains users **11-15** Reserved
102	66	CFISTRBP	1	binary	REBUILDPERCENT as specified in active CFRM policy
103	67	*	3	*	Reserved
106	6A	CFISTPL	35	EBCDIC	CF preference list
141	8D	CFISTXL	67	EBCDIC	Structure exclusion list
208	D0	CFISTETM	8	floating point	Summed structure execution time in microseconds
216	D8	CFISTSC1	4	binary	Index of first CFISSCM entry belonging to this structure
220	DC	CFISTMRC	4	binary	Number of times a request was found delayed due to coupling facility resource monopolization
224	E0	CFISTMTM	8	binary	Summed queue time (in microseconds) for operations queued due to coupling facility resource monopolization
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. The estimated maximum number of list elements that may reside in storage class memory is not included.
16	10	CFISTCUE	4	binary	List structure only: Current number of elements in use. The number of list elements that currently reside in storage class memory is not included.

Dec offset	Hex offset	Name	Length	Format	Description
20	14	CFISTLEL	4	binary	List structure only: Limit on number of list entries. The estimated maximum number of list entries that may reside in storage class memory is not included. Lock structure: Limit on number of data elements.
24	18	CFISTLEM	4	binary	List structure only: Current number of list entries used during MINTIME. The number of list entries that currently reside in storage class memory is not included. Lock structure: Current number of data elements in use.
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
72	48	CFISTXSS	4	binary	Maximum structure size (units = 4K byte blocks)
76	4C	CFISTMSS	4	binary	Minimum structure size (units = 4K byte blocks)
80	50	CFISTDTS	4	binary	Structure dump table size (4k blocks)
84	54	CFISTLHD	4	binary	Number of list headers (list only)
88	58	CFISTLEC	1	binary	List element characteristic. Size of list element in bytes is $256*(2**LELX)$
89	59	CFISTMDS	1	binary	Maximum data list entry size (maximum number of elements per entry)
90	5A	CFISTLTC	1	binary	Lock table entry characteristic
91	5B	CFISTLFL	1	binary	List structure flags
92	5C	CFISTDEC	1	binary	Data area element characteristic (cache only)
93	5D	CFISTDAS	1	binary	Maximum data area size (cache only)
94	5E	CFISTDSC	1	binary	Maximum storage class (cache only)
95	5F	*	1	*	Reserved
96	60	CFISTDCC	2	binary	Maximum castout class (cache only)
98	62	CFISTFCC	2	binary	First castout class (cache only)
100	64	CFISTLCC	2	binary	Last castout class (cache only)
102	66	CFISTCFL	1	binary	Cache structure flags
103	67	*	1	binary	Reserved
104	68	CFISTCEN	4	binary	Cache structure only: Total structure changed entry count.
108	6C	CFISTCEL	4	binary	Cache structure only: Total structure changed data element count.
CFICHPAS table entry section:					

ERBCFIG3 - Coupling facility data

Dec offset	Hex offset	Name	Length	Format	Description
0	0	CFICHPID	1	binary	ID of channel path connected to the CF
1	1	*	3	*	Reserved
4	4	CFICHEIN	2	binary	Index of according CFI table entry
6	6	*	2	*	Reserved
8	8	CFICHTYPE	1	binary	Channel path type
9	9	*	3	*	Reserved

12	C	CFICHFLAGS	4	binary	Channel path validity flags
					Bit
					    Meaning When Set
					0
					    HCA ID and port number valid
					1
					    Channel path operation mode valid
					2
					    Latency valid
					3
					    Degraded bit valid
					4-7
					    Corresponding entry in CFICHSAP valid
					8
					    Physical channel path ID valid
					9-31
					    Reserved

16	10	CFICHAID	2	binary	Host channel adapter ID (HCA)
18	12	CFICHAPN	1	binary	Host channel adapter port number
19	13	CFICHOPM	1	binary	Channel path operation mode
20	14	CFICHLAT	4	binary	Channel path latency time

24	18	CFICHSTA	1	binary	Channel path status flags
					Bit
					    Meaning When Set
					0
					    Channel path is operating in degraded mode
					1-7
					    Reserved

25	19	*	1	*	Reserved
26	1A	CFICHPCP	2	binary	Physical channel path ID
28	1C	CFICHSAP(4)	1	binary	System assist processor to which the channel path is accessible
32	20	*	4	*	Reserved
CFISSCMS table entry section:					
0	0	CFISCNAM	16	EBCDIC	Name of connected structure in this CF
16	10	CFISCVER	8	binary	Structure Version number
24	18	CFISCMAX	8	binary	Maximum amount of storage class memory the structure can use in 4K-blocks
32	20	CFISCALG	1	binary	SCM algorithm type
33	21	*	3	binary	Reserved
36	24	CFISCFAU	4	binary	Fixed augmented space in 4K-blocks

Dec offset	Hex offset	Name	Length	Format	Description
40	28	*	4	binary	Reserved
44	2C	CFISCIUA	4	binary	Amount of augmented space in use by the structure in 4K-blocks
48	30	CFISCIUS	8	binary	Amount of storage class memory in use by the structure in 4K-blocks
56	38	*	4	binary	Reserved
60	3C	CFISCEMA	4	binary	Estimated maximum amount of space that may be assigned as augmented space for the structure in 4K-blocks
64	40	CFISCEML	8	binary	Estimated maximum number of list entries that may reside in storage class memory
72	48	CFISCEME	8	binary	Estimated maximum number of list elements that may reside in storage class memory
80	50	CFISCENL	8	binary	Number of existing structure list entries that reside in storage class memory
88	58	CFISCENE	8	binary	Number of existing structure list elements that reside in storage class memory
CFICONNS table entry section:					
0	0	CFICNSIN	2	binary	Index of according CFISTRUS table entry
2	2	*	2	*	Reserved
4	4	CFICNSYS	8	EBCDIC	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

44	2C	CFICNFLG	2	binary	Connection flags
					Bit
					    Meaning When Set
					0
					    Rebuild allowed
					1
					    Duplexing rebuild allowed
					2
					    Structure alter allowed
					3
					    Auto allowed
					4
					    Auto allowed and suspend specified for connection. Valid only when ALLAuto=ON. Applicable only when CONN_Status = 1 .
					5-15
					    Reserved.

ERBCPCDB - CPC data control block

Dec offset	Hex offset	Name	Length	Format	Description
CPCDB header section:					
0	0	CPC_EyeCt	5	EBCDIC	Name of CPCDB
5	5	CPC_VerNum	1	binary	CPCDB version X'0C'
6	6	*	2	*	Reserved
8	8	CPC_HdrLen	4	binary	Length of CPCDB header
12	C	CPC_TotLen	4	binary	Total length of CPCDB
16	10	CPC_Flags	2	binary	Status flags **Bit** **Meaning when set** **0** No data from SYSEVENT REQLPDAT **1** Partition PHYSICAL exists **2** Data invalid **3** CPCDB converted to lower service level **4** CPCDB converted to higher release or service level **5–15** Reserved
18	12	CPC_MaxLpars	2	binary	Maximum number of LPARs
20	14	CPC_MaxProcs	2	binary	Maximum number of processors
22	16	CPC_PhysProcs	2	binary	Number of physical processors
24	18	CPC_Homeo	4	binary	Offset to home LPAR Section
28	1C	CPC_Homel	2	binary	Length of home LPAR Section
30	1E	CPC_LparMainL	2	binary	Length of CPC LPAR Section
32	20	CPC_LparO	4	binary	Offset to CPC LPAR Sections
36	24	CPC_LparL	2	binary	Length of CPC LPAR Section with CPC Logical Processor Section(s)
38	26	CPC_LparN	2	binary	Number of CPC LPAR Sections
40	28	CPC_DTime	8	EBCDIC	Time delta between two DIAG calls
Home LPAR section:					

Dec offset	Hex offset	Name	Length	Format	Description

0	0	CPC_HomeFlags	2	binary	Status flags
					Bit
					    Meaning when set
					0
					    Capacity values available
					1–2
					    Reserved
					3
					    VARYCPU option set
					4
					    WLM LPAR management enabled
					5
					    Multithreading measurements available
					6
					    ABSMSUCAPPING option set
					7–15
					    Reserved

2	2	CPC_BoostInfo	1	binary	Boost information
					Bit
					    Meaning when set
					0
					    zIIP boost active during entire MINTIME
					1
					    Speed boost active during entire MINTIME
					2–4
					    Reserved
					5–7
					    Boost class, valid only when zIIP boost and/or Speed boost is active
					        001: IPL
					        010: Shutdown
					        011: Recovery process

3	3	*	1	*	Reserved
4	4	CPC_CecMSU	4	binary	Effective processor capacity available to the CPC
8	8	CPC_LparMSU	4	binary	See LPDatImgCapacity of IRALPDAT
12	C	*	4	*	Reserved
16	10	CPC_HomeLPName	8	EBCDIC	Name of the home partition
24	18	CPC_PhysAdj	4	binary	See LPDatPhyCpuAdjFactor of IRALPDAT
28	1C	CPC_WeightCumD	4	binary	See LPDatCumWeight of IRALPDAT. This is the delta between begin and end of MINTIME.
32	20	CPC_WeightNumD	4	binary	See LPDatWeightAccumCounter of IRALPDAT. This is the delta between begin and end of MINTIME.
36	24	*	2	*	Reserved
38	26	CPC_CapAdj	1	binary	Capacity adjustment indication
39	27	CPC_CapRsn	1	binary	Capacity change reason
40	28	CPC_ImgMsuLimit	4	binary	Image capacity MSU limit
44	2C	CPC_4hAverage	4	binary	See LPDatAvgImgService of IRALPDAT
48	30	CPC_UncappedTimeD	8	binary	Uncapped time delta. See LPDatCumUncappedElapsedTime of IRALPDAT. This is the delta between begin and end of MINTIME.

Dec offset	Hex offset	Name	Length	Format	Description
56	38	CPC_CappedTimeD	8	binary	Capped time delta. See LPDatCumCappedElapsedTime of IRALPDAT. This is the delta between begin and end of MINTIME.
64	40	CPC_MsuInterval	4	binary	Approximate time interval (in seconds) for each entry in the MSU table. see LPDatServiceTableEntryInterval of IRALPDAT.
68	44	CPC_MsuDataEntries	4	binary	Number of WLM intervals within the last 4 hours.
72	48	*	384	*	Reserved
456	1C8	CPC_GrpCapName	8	EBCDIC	Name of the capacity group to which the partition belongs
464	1D0	CPC_GrpCapLimit	4	binary	MSU limit for the capacity group to which this partition belongs
468	1D4	*	4	*	Reserved
472	1D8	CPC_GrpJoinedTOD	8	binary	Time when this LPAR has joined the group (STCK format)
480	1E0	*	192	*	Reserved
672	2A0	CPC_Prod_AAP	4	binary	Multithreading core productivity numerator for AAP
676	2A4	CPC_Prod_IIP	4	binary	Multithreading core productivity numerator for IIP
680	2A8	CPC_Prod_CP	4	binary	Multithreading core productivity numerator for CP
684	2AC	CPC_MaxCapF_AAP	4	binary	Multithreading Maximum Capacity Factor numerator for AAP
688	2B0	CPC_MaxCapF_IIP	4	binary	Multithreading Maximum Capacity Factor numerator for IIP
692	2B4	CPC_MaxCapF_CP	4	binary	Multithreading Maximum Capacity Factor numerator for CP
696	2B8	CPC_CapF_AAP	4	binary	Multithreading Capacity Factor numerator for AAP
700	2BC	CPC_CapF_IIP	4	binary	Multithreading Capacity Factor numerator for IIP
704	2C0	CPC_CapF_CP	4	binary	Multithreading Capacity Factor numerator for CP
708	2C4	CPC_ATD_AAP	4	binary	Average Thread Density for AAP
712	2C8	CPC_ATD_IIP	4	binary	Average Thread Density for IIP
716	2CC	CPC_ATD_CP	4	binary	Average Thread Density for CP
720	2D0	CPC_MODE_AAP	2	binary	MT Mode AAP
722	2D2	CPC_MODE_IIP	2	binary	MT Mode IIP
724	2D4	CPC_MODE_CP	2	binary	MT Mode CP
726	2D6	*	14	binary	Reserved
740	2E4	CPC_CecName	8	EBCDIC	Name of the CPC
748	2EC	CPC_Time_To_Cap	2	binary	Remaining time until capping (in seconds)
750	2EE	CPC_Time_To_Cap_Group	2	binary	Remaining time until capacity group is subject to capping (in seconds)
752	2F0	CPC_zCBP_MCI	16	EBCDIC	Reserved
768	300	CPC_zCBP_MCR	4	binary	Reserved
772	304	CPC_zCBP_NomMCR	4	binary	Reserved
CPC LPAR section:					
0	0	CPC_LparName	8	EBCDIC	LPAR name
8	8	CPC_LparId	2	binary	LPAR number

Dec offset	Hex offset	Name	Length	Format	Description
10	A	CPC_LparFlags	1	binary	LPAR status flags **Bit** **Meaning when set** **0** This is the home partition **1** LPAR data invalid **2** Number of processors defined for this partition exceeds limit **3** CPC_UPID is valid **4** Partition belongs to a capacity group; CPC_GroupName and CPC_GroupMLU are valid **5** WLM weight management enabled **6** Initial weight instead of current weight should be used to project usage of the members in the capacity group. **7** Reserved
11	B	CPC_UPID	1	EBCDIC	User partition ID
12	C	CPC_LparDefMSU	4	binary	Defined MSU limit
16	10	CPC_OSname	8	EBCDIC	OS instance name
24	18	CPC_ProcO	4	binary	Offset to Logical Processor Sections
28	1C	CPC_ProcL	2	binary	Length of Logical Processor Section
30	1E	CPC_ProcN	2	binary	Number of Logical Processor Sections
32	20	CPC_LPCname	8	EBCDIC	LPAR cluster name
40	28	CPC_GroupName	8	EBCDIC	Name of the capacity group to which the partition belongs
48	30	CPC_GroupMLU	4	binary	Group maximum license units
52	34	CPC_OnlineCS	4	binary	Central storage (in MB) currently online to this partition
56	38	CPC_HWGroupName	8	EBCDIC	Name of hardware group to which this partition belongs
64	40	CPC_BoostType	1	binary	**Bit** **Meaning when set** **0** zIIP capacity boost active at end of MINTIME **1** Speed boost active at end of MINTIME **2–7** Reserved
65	41	*	7	*	Reserved
CPC logical processor section:					
0	0	CPC_ProcId	2	binary	Logical CPU address

Dec offset	Hex offset	Name	Length	Format	Description

2	2	CPC_ProcTyp	1	binary	Processor type:
					Value
					    Meaning
					1
					    CP
					2
					    ICF-pool
					3
					    AAP
					4
					    IFL
					5
					    ICF
					6
					    IIP

3	3	*	1	*	Reserved

4	4	CPC_ProcState	2	binary	Processor status indicators
					Bit
					    Meaning when set
					0
					    Processor not available in MINTIME
					1
					    Processor online
					2
					    Processor dedicated
					3
					    Wait completion = yes
					4
					    Wait completion = no
					5
					    Initial capping = 'ON'
					6
					    Polarization flag: this partition is vertically polarized; that is, HiperDispatch mode is active. CPC_ProcPolarWgt is valid.
					7–8
					    Polarization indicator:
					    00
					        Horizontally polarized or polarization not indicated
					    01
					        Vertically polarized with low entitlement
					    10
					        Vertically polarized with medium entitlement
					    11
					        Vertically polarized with high entitlement
					9–15
					    Reserved.

Dec offset	Hex offset	Name	Length	Format	Description

6	6	CPC_ProcChgInd	2	binary	Processor status-change indicators
					
					Bit
					    Meaning when set
					0
					    Changed from online to offline or vice versa
					1
					    Changed from shared to dedicated or vice versa
					2
					    'Initial capping' status changed
					3
					    Wait completion changed
					4
					    Maximum weight changed
					5
					    Absolute limit on partition usage changed
					6
					    Hardware group name or absolute limit on hardware group usage changed
					7
					    Topology has changed
					8–15
					    Reserved

8	8	CPC_ProcDispTimeD	8	binary	Dispatch time between begin and end of MINTIME in microseconds
16	10	CPC_ProcEffDispTimeD	8	binary	Effective dispatch time between begin and end of MINTIME in microseconds
24	18	CPC_ProcOnlineTimeD	8	binary	Online time between begin and end of MINTIME in microseconds
32	20	CPC_ProcMaxWeight	2	binary	Maximum LPAR share
34	22	CPC_ProcCurWeight	2	binary	Current LPAR share
36	24	CPC_ProcMinWeight	2	binary	Minimum LPAR share
38	26	CPC_ProcIniWeight	2	binary	Defined (initial) LPAR weight
40	28	CPC_ProcPolarWeight	4	binary	Weight for the logical CPU when HiperDispatch mode is active. See bit 6 of CPC_ProcState. Multiplied by a factor of 4096 for more granularity.
44	2C	CPC_HWCapLimit	4	binary	If not zero, absolute limit on partition usage of all CPUs of the type indicated in CPC_ProcTyp in terms of a number specified in hundredths of a CPU. For example, a value of 250 indicates that the partition is limited to using 2.5 CPUs.
48	30	CPC_HWGrpCapLimit	4	binary	If not zero, absolute limit on partition usage of all CPUs of the type indicated in CPC_ProcTyp that are members of the same hardware group, in terms of a number specified in hundredths of a CPU. For example, a value of 250 indicates that the hardware group is limited to using 2.5 CPUs.
52	34	CPC_ProcPMAWeight	4	binary	Pricing-management adjustment weight, aka phantom weight
Fields CPC_ProcMaxNL and CPC_ProcCordL1 - CPC_ProcCordL6 contain CPU topology location information for logical and physical core homes. This information is available if CPC_VerNum > X'0B'.					

Dec offset	Hex offset	Name	Length	Format	Description

56	38	CPC_ProcMaxNL	1	binary	Maximum number of topology nesting levels. The value is model dependent with a maximum of 6.
					Value
					    Meaning
					0
					    The model does not provide information about the topological nesting levels.
					1
					    There is no actual topological nesting structure.
					2–6
					    Topological nesting levels are available, beginning with field CPC_ProcCordL1 up to field CPC_ProcCordL*x*, where *x* is the value that defines the maximum number of topology nesting levels.

57	39	CPC_ProcCordL1	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 1. Valid if CPC_ProcMaxNL > 0.
58	3A	CPC_ProcCordL2	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 2. Valid if CPC_ProcMaxNL > 1.
59	3B	CPC_ProcCordL3	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 3. Valid if CPC_ProcMaxNL > 2.
60	3C	CPC_ProcCordL4	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 4. Valid if CPC_ProcMaxNL > 3.
61	3D	CPC_ProcCordL5	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 5. Valid if CPC_ProcMaxNL > 4.
62	3E	CPC_ProcCordL6	1	binary	Coordinate of the preferred dispatch location of the logical core at topological nesting level 6. Valid if CPC_ProcMaxNL > 5.
63	3F	*	9		Reserved

ERBCPDG3 - Channel data table

Dec offset	Hex offset	Name	Length	Format	Description
CPDG3 header section:					
0	0	CPDACR	5	EBCDIC	Acronym 'CPDG3'
5	5	CPDVER	1	binary	Control block version X'03'
6	6	CPDHDRL	2	binary	Length of CPDG3 header
8	8	CPDTOTL	4	binary	Total length of CPDG3
12	C	CPDNDAT	2	binary	Number of online channel path data sections
14	E	CPDLDAT	2	binary	Length of channel path data section
16	10	CPDODAT	4	binary	Offset to first channel path data section
Global channel information:					
20	14	CPDSMP	4	binary	Number of samples weighted by SRM, only valid if bit 6 of CPDFLG = OFF

Dec offset	Hex offset	Name	Length	Format	Description
24	18	CPDFLG	1	binary	Flags:
					Bit
					Meaning when set
					0
					No data
					1
					Running in LPAR mode
					2
					CPMF available
					3
					Configuration change
					4
					DCM supported by hardware
					5
					Configuration contains DCM managed channels
					6
					HW allows multiple channel subsystems
					7
					Reserved
25	19	CPDCMI	1	binary	CPMF mode info: 0 = CPMF not available 1 = Compatibility mode 2 = Extended mode
26	1A	CPDCSS	1	binary	Channel subsystem ID, only valid if bit 6 of CPDFLG = ON
27	1B	CPDLPN	1	binary	Logical partition number
28	1C	CPDCFRC	4	binary	CPMF restart count
32	20	CPDCFSC	4	binary	CPMF sample count
36	24	*	12	EBCDIC	For alignment
Channel path performance data:					
0	0	CPDCPID	1	EBCDIC	Channel path identification
1	1	CPDCFLG	1	binary	Channel flags
					Bit
					Meaning when set
					0
					Channel path is online.
					1
					Channel path is shared between logical partitions.
					2
					CPMF indication: this entry is invalid.
					3
					Channel path is DCM managed.
					4
					Channel characteristics changed.
					5
					Extended channel measurements are supported.
					6
					Channel path data utility string failure indicator.
					7
					Reserved.

ERBCPDG3 - Channel data table

Dec offset	Hex offset	Name	Length	Format	Description
2	2	CPDDFLG	1	binary	Channel data flags for channel measurement group 2 or 3 data **Bit** **Meaning when set** **0** Channel characteristics word 1 valid **1** Channel characteristics word 2 valid **2** Channel characteristics word 3 valid **3** Channel characteristics word 4 valid **4** Channel characteristics word 5 valid **5-7** Reserved
3	3	CPDCPD	1	binary	Channel path description. For an explanation, you can issue the command: D M=CHP.
4	4	CPDCPA	5	EBCDIC	Channel path acronym
9	9	CPDCMG	1	binary	Channel measurement group
10	A	CPDCPP	1	binary	Channel path parameter
11	B	CPDGEN	1	binary	Channel type generation
12	C	CPDBSY	4	binary	Number of samples the channel path was busy, weighted by SRM
16	10	CPDPBY	4	binary	LPAR channel-path-busy time in units of 128 micro-seconds
20	14	CPDCPTX	4	binary	Extended last time stamp
20	14	*	1	binary	Overflow area
21	15	CPDCPTS	3	binary	Last CPMB entry time stamp in units of 128 micro-seconds (note that this time value wraps about every 35.79 minutes)
24	18	*	24	*	Reserved
48	30	CPDCCMC	20	EBCDIC	Characteristics part
68	44	CPDCCMD	28	EBCDIC	Measurements part
96	60	CPDCCMX	32	EBCDIC	Extended channel measurement group data. Only valid if bit 5 of CPDCFLG = ON.
128	80	CPDPNETID1	16	EBCDIC	Physical-network identifier (PNET ID) of an Ethernet network that is accessible from the first port of the channel path. Only valid if bit 6 of CPDCFLG = OFF.
144	90	CPDPNETID2	16	EBCDIC	Physical-network identifier (PNET ID) of an Ethernet network that is accessible from the second port of the channel path. Only valid if bit 6 of CPDCFLG = OFF.
Channel measurement group 1 data - Characteristics part:					
0	0	CPDCC1	20	EBCDIC	not used
Channel measurement group 1 data - Measurements part:					
20	14	CPDTUT	4	floating point	CPMF total channel path busy time in units of 128 micro-seconds
24	18	CPDPUT	4	floating point	CPMF LPAR channel path busy time in units of 128 micro-seconds
28	1C	*	20	*	Reserved
Channel measurement group 2 data - Characteristics part:					

Dec offset	Hex offset	Name	Length	Format	Description
0	0	CPDMBC	4	floating point	CPMF maximum bus cycles per second (word 1)
4	4	CPDMCU	4	floating point	CPMF maximum channel work units per second (word 2)
8	8	CPDMWU	4	floating point	CPMF maximum write data units per second (word 3)
12	C	CPDMRU	4	floating point	CPMF maximum read data units per second (word 4)
16	10	CPDUS	4	floating point	CPMF data unit size in bytes (word 5)
Channel measurement group 2 data - Measurements part:					
20	14	CPDTBC	4	floating point	CPMF total bus cycle count
24	18	CPDTUC	4	floating point	CPMF total channel work unit count
28	1C	CPDPUC	4	floating point	CPMF LPAR channel work unit count
32	20	CPDTWU	4	floating point	CPMF total write data units
36	24	CPDPWU	4	floating point	CPMF LPAR write data units
40	28	CPDTRU	4	floating point	CPMF total read data units
44	2C	CPDPRU	4	floating point	CPMF LPAR read data units
Channel measurement group 2 data - Extended data:					
0	0	CPDTCO	4	floating point	Total number of FICON command-mode operations (CPC) that have been attempted by the channel.
4	4	CPDTCD	4	floating point	Total number of FICON command-mode operations (CPC) that could not be initiated by the channel due to a lack of available resources.
8	8	CPDTTO	4	floating point	Total number of FICON transport-mode operations (CPC) that have been attempted by the channel. Zero when no zHPF.
12	C	CPDTTD	4	floating point	Total number of FICON transport-mode operations (CPC) that could not be initiated by the channel due to a lack of available resources. Zero when no zHPF.
16	10	CPDTCS	8	floating point	Summation count of FICON command-mode operations (CPC). Each time the number of FICON command-mode operations is incremented, the number of FICON command-mode operations active at the channel, including the one being initiated, is added to this field.
24	18	CPDTTS	8	floating point	Summation count of FICON transport-mode operations (CPC). Each time the number of FICON transport-mode operations is incremented, the number of transport-mode operations active at the channel, including the one being initiated, is added to this field. Zero when no zHPF.
Channel measurement group 3 data - Characteristics part:					
0	0	CPDPDU	4	floating point	CPMF LPAR data unit size in bytes (word 1)
4	4	CPDTDU	4	floating point	CPMF total data unit size in bytes (word 2)

Dec offset	Hex offset	Name	Length	Format	Description
8	8	CPDPUM	4	floating point	CPMF LPAR message sent unit size (word 3)
12	C	CPDTUM	4	floating point	CPMF total message sent unit size (word 4)
16	10	*	4	*	Reserved
Channel measurement group 3 data - Measurements part:					
20	14	CPDPMS	4	floating point	CPMF LPAR message sent units count
24	18	CPDTMS	4	floating point	CPMF total message sent units count
28	1C	CPDPUS	4	floating point	CPMF LPAR count of unsuccessful attempts to send messages
32	20	CPDPUB	4	floating point	CPMF LPAR count of unsuccessful attempts to receive messages due to unavailable buffers
36	24	CPDTUB	4	floating point	CPMF total count of unsuccessful attempts to receive messages due to unavailable buffers
40	28	CPDPDS	4	floating point	CPMF LPAR data units sent count
44	2C	CPDTDS	4	floating point	CPMF total data units sent count

ERBCPUDB - CPU data block

Dec offset	Hex offset	Name	Length	Format	Description
0	0	CPUEYECT	5	EBCDIC	Acronym CPUDB
5	5	CPUVERNR	1	binary	CPUDB version X'04'
6	6	CPUHDRLN	2	binary	Length of CPU data block header
8	8	CPUTOTLN	4	binary	Length of CPU data block
12	C	CPUSUBPL	4	binary	Subpool
16	10	CPUENTLN	4	binary	Length of one processor entry
20	14	CPUENTNR	4	binary	Number of processor entries
24	18	CPUTIMST	8	binary	Time stamp when data was gathered, in TOD clock format
32	20	CPUTIMED	8	binary	Time range of (delta) data, in TOD clock format
40	28	CPUCCVUT	4	binary	SRM's view of CPU utilization (percentage), or 0 if not used
44	2C	CPUNUMCP	2	binary	Number of valid CPs
46	2E	CPUNUMOL	2	binary	Number of CPs online during interval (online at begin and end of interval)
48	30	CPUPARTN	8	EBCDIC	Partition name

Dec offset	Hex offset	Name	Length	Format	Description

56	38	CPUFLAGS	4	binary	Flags
					
					Bit
					    Meaning when set
					0
					    BASIC mode
					1
					    LPAR mode
					2
					    Diagnose X'204' failure if in LPAR mode
					3
					    Configuration change
					4
					    Partition name changed if in LPAR mode
					5
					    Number of valid CPs changed
					6
					    Diagnose X'204' returns extended data
					7
					    PROCVIEW CORE active
					8
					    HISMT failure
					9
					    CPUDB converted to lower service level
					10
					    CPUDB converted to higher release or service level
					11–31
					    Reserved

60	3C	CPUNIFA	2	binary	Number of valid zAAPs
62	3E	CPUNSUP	2	binary	Number of valid zIIPs
64	40	*	64	*	Reserved
CPU extension for each processor:					
0	0	CPUID	2	binary	Processor ID

ERBCPUDB - CPU data block

Dec offset	Hex offset	Name	Length	Format	Description
2	2	CPUSTATE	2	binary	CPU state indication
					Bit
					Meaning when set
					0
					Valid data.
					1
					Only partial.
					2
					Alive at end of interval.
					3
					Status changed.
					4
					PR/SM view of this processor changed.
					5
					Processor is dedicated.
					6
					Dedicated status changed.
					7
					Wait completion = YES.
					8
					Wait completion status changed.
					9
					Capping partition flag. Capped = YES in logical partition controls.
					10
					Capping partition status changed.
					11
					Processing weight changed.
					12
					Multi-threading Core LPAR Busy time is valid.
					13
					CPU is offline while core is online.
					14–15
					Reserved.
4	4	CPUWEIGH	2	binary	The processing weight assigned to this partition. This value is in the range 0 to 999, except that it is 65535 if the processor is dedicated. Only if LPAR mode.
6	6	CPUPRTYP	1	binary	CPU type
					Value
					Meaning
					0
					CP
					1
					zAAP
					2
					zIIP
7	7	*	1	*	Reserved
8	8	CPUDISPT	8	binary	Total dispatch time, in microseconds
16	10	CPUEFDTT	8	binary	Total effective dispatch time, in microseconds
24	18	CPUWAITT	8	binary	Total wait time, in microseconds
32	20	CPUONLTT	8	binary	Total online time, in microseconds
40	28	CPUPARKT	8	binary	Total parked time, in TOD clock format
48	30	CPULPBTT	4	binary	Total Multi-Threading Core LPAR Busy time, in milliseconds

Dec offset	Hex offset	Name	Length	Format	Description
52	34	CPUCID	2	binary	Logical core ID
54	36	CPUTID	2	binary	Thread ID. Can be a non-zero value if multi-threading mode > 1.

ERBCPUG3 - Processor data control block

Dec offset	Hex offset	Name	Length	Format	Description
0	0	CPUG3_AC	5	EBCDIC	Name of CPUG3
5	5	CPUG3_VE	1	binary	CPUG3 version X'06'
6	6	*	1	*	Reserved

7	7	CPUG3_FLAG	1	binary	CPUG3 flags
					
					Bit
					    Meaning when set
					0
					    CPUG3 converted to lower service level
					1
					    CPUG3 converted to higher release or service level
					2 - 7
					    Reserved

8	8	CPUG3_HDRL	4	binary	Length of CPUG3 header
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 cores. In the case of a native (non PR/SM) system, this time is equal to the logical CPU time

Dec offset	Hex offset	Name	Length	Format	Description

40	28	CPUG3_STATUS	4	binary	Status information
					
					Bit
					    Meaning When Set
					0
					    BASIC mode system
					1
					    LPAR mode system
					2
					    Running as a z/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
					    HISMT failed
					10
					    Running under an alternate virtual machine
					11-31
					    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	CPUG3_CPCOFF	4	binary	Offset to CPCDB area
60	3C	CPUG3_IFCON	4	binary	Number of zAAPs online at end of range
64	40	CPUG3_NUMIFCOL	4	binary	Accumulated number of zAAPs online. To get average number, divide by number of samples.
68	44	CPUG3_NUMPRIFA	8	binary	Accumulated online time of zAAPs, in microseconds
76	4C	CPUG3_LOGITIFA	8	binary	Logical CPU time: The sum of MVS NON_WAIT time of all online logical zAAPs in the time range, in microseconds.
84	54	CPUG3_PHYSTIFA	8	binary	Physical CPU time: The sum of all CPU times used by all online logical zAAPs in the time range (in microseconds).
92	5C	CPUG3_SUCON	4	binary	Number of zIIPs online at end of range
96	60	CPUG3_NUMSUCOL	4	binary	Accumulated number of zIIPs online. To get average number, divide by number of samples.
100	64	CPUG3_NUMPRSUP	8	binary	Accumulated online time of zIIPs, in microseconds
108	6C	CPUG3_LOGITSUP	8	binary	Logical CPU time: The sum of MVS NON_WAIT time of all online logical zIIPs in the time range (in microseconds)
116	74	CPUG3_PHYSTSUP	8	binary	Physical CPU time: The sum of all CPU times used by all online logical zIIPs in the time range (in microseconds)
124	7C	CPUG3_PARK_CP	8	binary	Accumulated parked time on CPs, in microseconds

Dec offset	Hex offset	Name	Length	Format	Description
132	84	CPUG3_PARK_IFA	8	binary	Accumulated parked time on zAAPs, in microseconds
140	8C	CPUG3_PARK_SUP	8	binary	Accumulated parked time on zIIPs, in microseconds
148	94	CPUG3_CPUOFF	4	binary	Offset to CPUDB data area
152	98	CPUG3_CPOnlCore#	4	binary	Accumulated number of online CP cores. To get average number divide by number of samples.
156	9C	CPUG3_IFAOnlCore#	4	binary	Accumulated number of online zAAP cores. To get average number divide by number of samples.
160	A0	CPUG3_SUPOnlCore#	4	binary	Accumulated number of online zIIP cores. To get average number divide by number of samples.
164	A4	*	12	*	Reserved

ERBCRYG3 - Cryptographic hardware data table

Dec offset	Hex offset	Name	Length	Format	Description
CRYG3 header section:					
0	0	CRYEyeC	5	EBCDIC	Eyecatcher CRYG3
5	5	CRYVer	1	binary	CRYG3 version
6	6	CRYCycbRc	2	binary	Return code from CHSC
8	8	CRYHdrL	4	binary	Length of CRYG3 header
12	C	CRYTotL	4	binary	Total length of CRYG3
16	10	CRYCpcN	8	EBCDIC	CPC name
24	18	CRYSysN	8	EBCDIC	System name
32	20	CRYSmfID	4	EBCDIC	SMF system ID
36	24	CRYMDatL	2	binary	Length of CRYMD entry
38	26	CRYMDatN	2	binary	Number of CRYMD entries
40	28	CRYMDatO	4	binary	Offset to CRYMD entries
CRYMD table entry section:					
0	0	CRYMD_EyeC	5	EBCDIC	Eyecatcher CRYMD
5	5	CRYMD_Ver	1	binary	CRYMD version
6	6	*	2	*	Reserved
8	8	APIndex	1	binary	Crypto processor index

9	9	APType	1	binary	Crypto processor type
					Value
					    Meaning
					11
					    CEX5S
					12
					    CEX6S
					13
					    CEX7S
					14
					    CEX8S

Dec offset	Hex offset	Name	Length	Format	Description

10	A	MapType	1	binary	Mapping type
					
					Value
					    Meaning
					8
					    Accelerator mode
					9
					    Coprocessor mode (CCA)
					10
					    XCP mode (PKCS11)

11	B	*	1	*	Reserved

12	C	Mask	4	binary	Validity bit mask.
					
					Bit
					    Meaning when set
					0
					    Timer-counter pair 0 is valid.
					1
					    Timer-counter pair 1 is valid.
					2
					    Timer-counter pair 2 is valid.
					3
					    Timer-counter pair 3 is valid.
					4
					    Timer-counter pair 4 is valid.
					5
					    Timer-counter pair 5 is valid.
					6–31
					    Reserved

16	10	ScFactor	8	floating point	Scaling factor for the indicated crypto type. Execution times must be multiplied by this scaling factor to achieve a value in seconds.
24	18	TC#	4	binary	Number of valid entries in the TC array of timer and counter pairs
28	1C	*	4	*	Reserved
32	20	TC(0-5)	*	*	Array with timer-counter pairs for the crypto function.
32	20	Time	8	binary	Execution time
40	28	NumOp	8	binary	Number of operations

128	80	Scope	1	binary	Specifies the scope of the data section:
					
					Value
					    Meaning
					0
					    Data with CPC scope
					1
					    Data with system scope

129	81	DomainID	1	binary	Domain ID, valid with Scope = 1
130	82	*	6	*	Reserved

ERBCSRG3 - Common storage remaining table

Dec offset	Hex offset	Name	Length	Format	Description
CSRG3 header section:					

Dec offset	Hex offset	Name	Length	Format	Description
0	0	CSRCSRG3	5	EBCDIC	Acronym 'CSRG3'
5	5	CSRVERG3	1	binary	Control block version X'02'
6	6	*	2	*	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	Job name
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 when set
					0
					    CSA and RUCSA amounts incomplete.
					1
					    SQA amounts incomplete.
					2–15
					    Reserved.

46	2E	*	2	*	Reserved
48	30	CSRRUCSA	4	binary	RUCSA amount
52	34	CSRERUCSA	4	binary	ERUCSA amount

ERBDDNG3 - Device data set name list table

Dec offset	Hex offset	Name	Length	Format	Description
Data set name list table header:					
0	0	DDNNAME	5	EBCDIC	Acronym 'DDNG3'
5	5	DDNVER	1	binary	DDNG3 version X'02'
6	6	DDNHLN	2	binary	Common header length
8	8	*	4	*	Reserved
12	C	DDNGMLN	4	binary	Allocated length of data set name list table

ERBDSIG3 - DS header

Dec offset	Hex offset	Name	Length	Format	Description
16	10	DDNMXEN	4	binary	Maximum number of table entries
20	14	DDNIOFS	4	binary	Offset to data set name index table
24	18	DDNILN	4	binary	Length of data set name index table
28	1C	DDNOOFS	4	binary	Offset to data set name offset table
32	20	DDNOLN	4	binary	Length of data set name offset table
36	24	DDNLOFS	4	binary	Offset to data set name list table
40	28	DDNLLN	4	binary	Length of data set name list table
44	2C	DDNCREN	4	binary	Current entry number
Data set name index table:					
0	0	DDNINAME	4	EBCDIC	Acronym 'DDNI'
4	4	DDNIAR(*)	*	*	Array of data set name indices
Data set name index entry:					
0	0	DDNIEL	4	binary	Data set name index
Data set name offset table:					
0	0	DDNONAME	4	EBCDIC	Acronym 'DDNO'
4	4	DDNOAR(*)	*	*	Array of data set name offsets
Data set name offset entry:					
0	0	DDNOEL	4	binary	Data set name offset
Data set name list table:					
0	0	DDNLNAME	4	EBCDIC	Acronym 'DDNL'
4	4	DDNLLST	0	*	Begin of data set name list
Data set name list entry:					
0	0	DDNLELN	2	binary	Length of data set name including 3 bytes for device number and subchannel set ID
2	2	DDNLENM	*	EBCDIC	Data set name including 3 bytes for device number and subchannel set ID

ERBDSIG3 - Data set header and index

Dec offset	Hex offset	Name	Length	Format	Description
DSIG3 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 opened
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

Dec offset	Hex offset	Name	Length	Format	Description
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
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

Dec offset	Hex offset	Name	Length	Format	Description
DVTG3 header section:					
0	0	DVTDVTG3	5	EBCDIC	Acronym 'DVTG3'
5	5	DVTVERG3	1	binary	Control block version X'12'
6	6	DVTHDRLE	1	binary	Length of 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

Dec offset	Hex offset	Name	Length	Format	Description
16	10	DVTENTRY	152	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	Reserved
8	8	DVTDEVNR	2	binary	Device number in hexadecimal format
10	A	DVTPREVI	2	binary	Reserved
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
24	18	DVTDISIF	4	binary	Native device DISC time at the beginning of the MINTIME for this set of samples (in 2048-microsecond units)

Dec offset	Hex offset	Name	Length	Format	Description
28	1C	DVTPETIF	4	binary	Native device PEND time at the beginning of the MINTIME for this set of samples (in 2048-microsecond units)
32	20	DVTCOTIF	4	binary	Native device CONN time at the beginning of the MINTIME for this set of samples (in 2048-microsecond units)
36	24	DVTDVBIF	4	binary	Device busy delay time at the beginning of the MINTIME for this set of samples (in 2048-microsecond units)
40	28	DVTCUBIF	4	*	No longer used
44	2C	DVTDISIL	4	binary	Native device DISC time at the end of the MINTIME for this set of samples (in 2048-microsecond units)
48	30	DVTPETIL	4	binary	Native device PEND time at the end of the MINTIME for this set of samples (in 2048-microsecond units)
52	34	DVTCOTIL	4	binary	Native device CONN time at the end of the MINTIME for this set of samples (in 2048-microsecond units)
56	38	DVTDVBIL	4	binary	Device busy delay time at the end of the MINTIME for this set of samples (in 2048-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
116	74	DVTCUQTP	4	binary	Control unit queuing time previous sample
120	78	DVTCUQTN	4	binary	Accumulated control unit queuing time for devices not connected to FICON channel
124	7C	DVTCUQTF	4	binary	Accumulated control unit queuing time for devices connected to FICON channel
128	80	DVTHPNUM	4	signed	Accumulated number of HyperPAV aliases in each cycle
132	84	DVTPSM	4	signed	Number of successful PAV samples

136	88	DVTFLAG3	1	binary	Flag byte
					
					Bit
					    Meaning when set
					0
					    Device is a HyperPAV base device
					1
					    HyperWrite requested on this device
					2–7
					    Reserved

137	89	DVTHPCON	1	binary	Configured HyperPAV aliases for that LSS
138	8A	*	3	*	Reserved

Dec offset	Hex offset	Name	Length	Format	Description
141	8D	DVTSSID	1	binary	Subchannel set
142	8E	DVTDEVN2	2	binary	Device number same as DVTDEVNR
144	90	DVTENIDX4	4	binary	Index of this DVTG3 entry
148	94	DVTPREVI4	4	binary	Index of previous DVTG3 entry

ERBENCG3 - Enclave data table

Dec offset	Hex offset	Name	Length	Format	Description
ENCARRAY:					
0	0	ENCG3ACR	5	EBCDIC	Acronym 'ENCG3'
5	5	ENCG3VER	1	binary	Control block version X'07'
6	6	*	2	*	Reserved
8	8	ENCG3TLN	4	binary	ENCG3 table length
12	C	ENCG3TET (6)	12	binary	Table entry triplets
12	C	ENCG3TEO	4	binary	Table entry offset
16	10	ENCG3TEL	4	binary	Table entry length
20	14	ENCG3TEN	4	binary	Table entry number
84	54	ENCG3DEO	4	binary	Descriptor entry offset
88	58	ENCG3DEL	4	binary	Descriptor entry length
92	5C	ENCG3DEN	4	binary	Descriptor entry number
ENCG3 header section:					
0	0	ENCG3LEN	4	binary	Table entry length
4	4	ENCTOKEN	8	EBCDIC	Enclave token
12	C	ENCCLX	2	binary	Service class index
12	C	ENCPGN	2	binary	Performance group
14	E	ENCSRPG	2	binary	Subsystem RCLX/RPGN
16	10	ENCNRPG	2	binary	Trx name RPGN
18	12	ENCURPG	2	binary	User ID RPGN
20	14	ENCCRPG	2	binary	Trx class RPGN
22	16	ENCARPG	2	binary	Account no RPGN
24	18	ENCPER	1	binary	SC|PG period
25	19	ENCDMN	1	binary	Domain
26	1A	ENCG3KFI	1	binary	Key field status flags **Bit** **Meaning when set** **0** Key SC/PG has changed. **1** Key period has changed. **2** Domain has changed. **3–7** Reserved.

Dec offset	Hex offset	Name	Length	Format	Description
27	1B	*	9	*	Reserved
36	24	ENCG3EDO	4	binary	Offset from ENCG3 element to EDEG3 element
40	28	ENCG3SMP	4	binary	Sample count
44	2C	ENCUSTOT	4	binary	Using count total
48	30	ENCDETOT	4	binary	Delay count total
52	34	ENCIDLES	4	binary	IDLE sample counts
56	38	ENCUNKNS	4	binary	UNKNOWN sample counts
60	3C	ENCUSCPU	4	binary	Using count CPU
64	40	ENCDECPU	4	binary	Delay count CPU
68	44	ENCDECCA	4	binary	Delay count CPU capping
72	48	ENCDESTG	4	binary	Delay count STOR paging
76	4C	ENCDECOM	4	binary	Delay count COM paging
80	50	ENCDEXMM	4	binary	Delay count X/M
84	54	ENCDESHP	4	binary	Delay count Shared pag
88	58	ENCFLAGS	2	binary	ENCG3 descriptive flags **Bit** **Meaning when set** **0** dependent enclave **1** original independent enclave **2** foreign independent enclave **3** foreign dependent enclave **4–15** Reserved
90	5A	ENCOASID	2	binary	Owner ASID
92	5C	ENCTOTS	4	binary	Multistate samples
96	60	ENCUMCPU	4	binary	Using count CPU (multistate samples)
100	64	ENCUMIO	4	binary	Using count I/O
104	68	ENCDMCPU	4	binary	Delay count CPU (multistate samples)
108	6C	ENCDMIO	4	binary	Delay count I/O
112	70	ENCDMQUE	4	binary	Delay count queue
116	74	ENCDMCCA	4	binary	Delay count capping
120	78	ENCDMSTO	4	binary	Delay count storage
124	7C	ENCMIDLE	4	binary	Idle count
128	80	ENCMUNKN	4	binary	Unknown count
132	84	ENCTCPUT	4	floating point	CPU time since creation of enclave
136	88	ENCCPUT	4	floating point	CPU time
140	8C	*	8	*	Reserved
148	94	ENCOWSYS	8	EBCDIC	Enclave owner system or blank if not a foreign enclave

ERBENCG3 - Enclave table

Dec offset	Hex offset	Name	Length	Format	Description
156	9C	ENCOWJOB	8	EBCDIC	Enclave owner job name or blank if not a foreign enclave
164	9C	ENCXTOK	32	EBCDIC	Enclave export token or zero if not a multi-system enclave
196	C4	ENCTIFAT	4	floating point	zAAP time since creation of enclave
200	C8	ENCTIFCT	4	floating point	zAAP on CP time since creation of enclave
204	CC	ENCIFAT	4	floating point	zAAP time
208	D0	ENCIFCT	4	floating point	zAAP on CP time
228	E4	ENCUSIFA	4	binary	Using count zAAP
232	E8	ENCUSIFC	4	binary	Using count zAAP on CP
236	EC	ENCDEIFA	4	binary	Delay count zAAP
240	F0	ENCUMIFA	4	binary	Using count zAAP (multistate samples)
244	F4	ENCUMIFC	4	binary	Using count zAAP on CP (multistate samples)
248	F8	ENCDMIFA	4	binary	Delay count zAAP (multistate samples)
252	FC	ENCUSCP	4	binary	Using count CP (single state samples)
256	100	ENCDECP	4	binary	Delay count CP (single state samples)
260	104	ENCTSUPT	4	floating point	zIIP time since creation of enclave
264	108	ENCTSUCT	4	floating point	zIIP on CP time since creation of enclave
268	10C	ENCSUPT	4	floating point	zIIP time
272	110	ENCSUCT	4	floating point	zIIP on CP time
276	114	*	16	*	Reserved
292	124	ENCUSSUP	4	binary	Using count zIIP (single state sample)
296	128	ENCUSSUC	4	binary	Using count zIIP on CP (single state sample)
300	12C	ENCDESUP	4	binary	Delay count zIIP (single state sample)
304	130	ENCUMSUP	4	binary	Using count zIIP (multi state sample)
308	134	ENCUMSUC	4	binary	Using count zIIP on CP (multi state sample)
312	138	ENCDMSUP	4	binary	Delay count zIIP (multi state sample)
Enclave descriptor entry (EDEG3):					
0	0	EDETRXN	8	EBCDIC	Transaction program name
8	8	EDEUSER	8	EBCDIC	User ID
16	10	EDETRXC	8	EBCDIC	Transaction class
24	18	EDENET	8	EBCDIC	Network ID
32	20	EDELU	8	EBCDIC	Logical unit name
40	28	EDEPLAN	8	EBCDIC	Plan
48	30	EDEPCKG	8	EBCDIC	Package name (filled if EDE_PackageNameLong has not been filled)

Dec offset	Hex offset	Name	Length	Format	Description
56	38	EDECNCTN	8	EBCDIC	Connection
64	40	EDECOLL	18	EBCDIC	Collection
82	52	EDECORR	12	EBCDIC	Correlation
94	5E	ECDSUBT	4	EBCDIC	Subsystem type
98	62	ECDFCN	8	EBCDIC	Function name
106	6A	ECDSUBN	8	EBCDIC	Subsystem name
114	72	EDESSPM	255	EBCDIC	Subsystem parameter
369	171	EDEACCT	143	EBCDIC	Accounting info
512	200	EDE_ PROCEDURENAME	18	EBCDIC	Procedure name (filled if EDE_ProcedureNameLong has not been filled)
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	EDE_ SchedulingEnvironment	16	EBCDIC	Scheduling environment
592	250	EDE_ SchedulingEnvironment_ Len	1	EBCDIC	Length of EDE_SchedulingEnvironment
593	251	*	3	*	Reserved
596	254	EDE_ SubsystemCollectionName	8	EBCDIC	Subsystem collection
604	25C	EDE_PackageNameLong	128	EBCDIC	Package name - long version
732	2DC	EDE_ PackageNameLong_Len	2	binary	Length of EDE_PackageNameLong
734	2DE	EDE_ProcedureNameLong	128	EBCDIC	Procedure name - long version
862	35E	EDE_ ProcedureNameLong_Len	2	binary	Length of EDE_ProcedureNameLong
864	360	EDE_ClientIPAddress	39	EBCDIC	Client IP address
903	387	EDE_ClientIPAddress_Len	1	binary	Length of EDE_ClientIPAddress
904	388	EDE_ClientUserID	128	EBCDIC	Client user ID
1032	408	EDE_ClientUserID_Len	2	binary	Length of EDE_ClientUserID
1034	40A	EDE_ClientTrxName	255	EBCDIC	Client transaction name
1289	509	*	1	*	Reserved
1290	50A	EDE_ClientTrxName_Len	2	binary	Length of EDE_ClientTrxName
1292	50C	EDE_ClientWksName	255	EBCDIC	Client workstation or hostname
1547	60B	*	1	*	Reserved
1548	60C	EDE_ClientWksName_Len	2	binary	Length of EDE_EDE_ClientWksName
1550	60E	EDE_ClientAccounting	512	EBCDIC	Client accounting information
2062	80E	EDE_ClientAccounting_Len	2	binary	Length of EDE_ClientAccounting

ERBENTG3 - Enqueue name table

Dec offset	Hex offset	Name	Length	Format	Description
ENTG3 header section:					
0	0	ENTENTG3	5	EBCDIC	Acronym 'ENTG3'

Dec offset	Hex offset	Name	Length	Format	Description
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

Dec Offset	Hex Offset	Name	Length	Format	Description
0	0	GEIGEIG3	5	EBCDIC	Acronym 'GEIG3'
5	5	GEIVERG3	1	binary	GEIG3 version X'19'
6	6	GEILEN	2	binary	Length of this control block (GEIG3)
8	8	*	16	*	Reserved
24	18	GEIVERSN	1	binary	CPU version number

Dec Offset	Hex Offset	Name	Length	Format	Description
25	19	GEIFLG3	1	binary	GEIG3 flags
					Bit
					Meaning when set
					0
					GEIG3 converted to lower service level
					1
					GEIG3 converted to higher release or service level
					2 - 7
					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 GRB3GGSS terminated
					7
					No ENQ contention data available due to GRS system problem
27	1B	GEIFLG1	1	binary	Additional flags
					Bit
					Meaning When Set
					0
					No ENQ contention data available because of GRS interface problem.
					1
					z/Architecture mode.
					2
					CMR data available.
					3
					zAAPs available.
					4
					zIIPs available.
					5
					Enhanced DAT facility 1 available.
					6
					Pageable large pages support enabled.
					7
					At least one zIIP is currently installed.
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

ERBGEIG3 - General table

Dec Offset	Hex Offset	Name	Length	Format	Description
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_VE	4	floating point	Current system UIC[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	GEIRSHR_VE	4	floating point	Number of real storage shared frames[1]
72	48	*	4	*	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]
92	5C	*	4	*	Reserved
96	60	GEIESQAF_VE	4	floating point	Reserved
100	64	GEIRLPAF_VE	4	floating point	Number of central storage LPA frames[1]
104	68	GEIELPAF_VE	4	floating point	Reserved
108	6C	GEIRCSAF_VE	4	floating point	Number of central storage CSA frames[1]
112	70	GEIECSAF_VE	4	floating point	Reserved
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[1]
124	7C	GEICSARE	4	binary	Amount of unallocated common area left (CSA + SQA)
128	80	*	4	*	Reserved
132	84	GEICPUON	2	binary	Snapshot number of online processors at end of the MINTIME[1]
134	86	*	2	*	Reserved
136	88	GEICSASZ	4	binary	IPL Size of CSA below 16M
140	8C	GEISQASZ	4	binary	IPL Size of SQA below 16M

Dec Offset	Hex Offset	Name	Length	Format	Description
144	90	GEIECSAZ	4	binary	IPL Size of CSA above 16M
148	94	GEIESQAZ	4	binary	IPL Size of SQA above 16M
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 16M
180	B4	GEISQAMX	4	binary	Max. allocated SQA below 16M
184	B8	GEIECSAX	4	binary	Max. allocated CSA above 16M
188	BC	GEIESQAX	4	binary	Max. allocated SQA above 16M
192	C0	GEICSASP	4	binary	Current allocated CSA below 16M
196	C4	GEISQASP	4	binary	Current allocated SQA below 16M
200	C8	GEIECSAP	4	binary	Current allocated CSA above 16M
204	CC	GEIESQAP	4	binary	Current allocated SQA above 16M
208	D0	GEICSAAV	4	floating point	Accumulated allocated CSA below 16M[1]
212	D4	GEISQAAV	4	floating point	Accumulated allocated SQA below 16M[1]
216	D8	GEIECSAV	4	floating point	Accumulated allocated CSA above 16M[1]
220	DC	GEIESQAV	4	floating point	Accumulated allocated SQA above 16M[1]
224	E0	GEICSACN	4	floating point	Accumulated CSA conv. below 16M[1]
228	E4	GEIECSAN	4	floating point	Accumulated CSA conv. above 16M[1]
232	E8	GEICSACE	4	binary	Snapshot CSA conv. below 16M
236	EC	GEIECSAE	4	binary	Snapshot CSA conv. above 16M
240	F0	GEICSAAS	4	floating point	Accumulated allocated CSA below 16M (held by the system)[1]
244	F4	GEISQAAS	4	floating point	Accumulated allocated SQA below 16M (held by the system)[1]
248	F8	GEIECSAS	4	floating point	Accumulated allocated CSA above 16M (held by the system)[1]
252	FC	GEIESQAS	4	floating point	Accumulated allocated SQA above 16M (held by the system)[1]
256	100	GEIBATCS	4	floating point	Accumulated allocated CSA below 16M (held by BATCH initiators)[1]
260	104	GEIBATEC	4	floating point	Accumulated allocated CSA above 16M (held by BATCH initiators)[1]
264	108	GEIBATSQ	4	floating point	Accumulated allocated SQA below 16M (held by BATCH initiators)[1]
268	10C	GEIBATES	4	floating point	Accumulated allocated SQA above 16M (held by BATCH initiators)[1]

Dec Offset	Hex Offset	Name	Length	Format	Description
272	110	GEIASCCS	4	floating point	Accumulated allocated CSA below 16M (held by ASCH initiators)[1]
276	114	GEIASCEC	4	floating point	Accumulated allocated CSA above 16M (held by ASCH initiators)[1]
280	118	GEIASCSQ	4	floating point	Accumulated allocated SQA below 16M (held by ASCH initiators)[1]
284	11C	GEIASCES	4	floating point	Accumulated allocated SQA above 16M (held by ASCH initiators)[1]
288	120	GEIOMVCS	4	floating point	Accumulated allocated CSA below 16M (held by OMVS initiators)[1]
292	124	GEIOMVEC	4	floating point	Accumulated allocated CSA above 16M (held by OMVS initiators)[1]
296	128	GEIOMVSQ	4	floating point	Accumulated allocated SQA below 16M (held by OMVS initiators)[1]
300	12C	GEIOMVES	4	floating point	Accumulated allocated SQA above 16M (held by OMVS initiators)[1]
304	130	GEIMTFLG	1	binary	Indicators for the current MINTIME **Bit** **Meaning When Set** **0** IPS changed during this MINTIME **1** CSA and RUCSA 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	GEIFLG2	1	binary	Additional flags **Bit** **Meaning When Set** **0** Enhanced DAT facility 2 available. **1** Reserved. **2** Restricted use common service area (RUCSA) is defined. **3-7** Reserved.
306	132	*	2	*	Reserved
308	134	GEISLID	4	EBCDIC	ID of slip trap
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

Dec Offset	Hex Offset	Name	Length	Format	Description
344	158	GEIMAXAS	4	binary	Maximum number of address spaces
348	15C	GEIESPMB	4	floating point	Reserved
352	160	GEIESPME	4	floating point	Reserved
356	164	GEIESMRB	4	floating point	Reserved
360	168	GEIESMRE	4	floating point	Reserved
364	16C	GEIMDL	16	EBCDIC	CPC model identifier
380	17C	GEISEQ	16	EBCDIC	CPC sequence number
396	18C	GEILOAL	4	floating point	User region value allocated below 16M[1]
400	190	GEIHIAL	4	floating point	LSQA/SWA/229/230 value allocated below 16M[1]
404	194	GEIELOAL	4	floating point	User region value allocated above 16M[1]
408	198	GEIEHIAL	4	floating point	LSQA/SWA/229/230 value allocated above 16M[1]
412	19C	GEITOTPI	4	floating point	Total number of paged-in pages, excluding swap-in, VIO, and hiperspace page-ins
416	1A0	GEISLTA	4	floating point	Number of currently available slots [1]
420	1A4	GEIRLMO	4	floating point	Number of fixed memory objects allocated in the system that can be backed in 1 MB frames [1]
424	1A8	GEIRLPR	4	floating point	Number of 1 MB pages fixed in central storage [1]
428	1AC	GEICMO	4	floating point	Number of high virtual common memory objects allocated [1]
432	1B0	GEICFR	4	floating point	Number of high virtual common memory pages backed in central storage [1]
436	1B4	GEICFFR	4	floating point	Number of high virtual common memory pages fixed in central storage [1]
440	1B8	GEICASL	4	floating point	Number of high virtual common memory auxiliary storage slots (DASD and SCM) [1]
444	1BC	GEISMO	4	floating point	Number of high virtual shared memory objects allocated [1]
448	1C0	GEISFR	4	floating point	Number of high virtual shared memory pages backed in central storage [1]
452	1C4	GEICSIZ	8	floating point	High virtual common area size
460	1CC	GEISSIZ	8	floating point	High virtual shared area size[1]
468	1D4	GEILSIZ	4	floating point	Maximum number of 1 MB frames that can be used by fixed 1 MB pages[1]
472	1D8	GEIRTFIX	4	floating point	Total number of fixed pages[1]
476	1DC	GEIRBFIX	4	floating point	Number of fixed frames below 16 MB central storage[1]
480	1E0	*	4	*	Reserved

Dec Offset	Hex Offset	Name	Length	Format	Description
484	1E4	GEICUSE	8	floating point	Number of high virtual common pages in-use
492	1EC	GEISUSE	8	floating point	Number of pages in use by shared memory objects.
500	1F4	GEILPAG	4	floating point	Reserved
504	1F8	GEILFUSE	4	floating point	Number of 1 MB frames that are in-use and are no longer available for fixed 1 MB pages.
508	1FC	GEILPUSE	4	floating point	Number of 1 MB frames used by pageable/DREF memory objects.
512	200	GEISASL	4	floating point	Number of high virtual shared memory auxiliary storage slots (DASD and SCM)[1]
516	204	GEIRSTRF	8	floating point	Number of online real storage frames[1]
524	20C	GEILCPR	8	floating point	Number of 1 MB high virtual common memory pages backed in central storage [1]
532	214	GEILCMO	4	floating point	Number of fixed memory objects allocated in common storage that can be backed in 1 MB frames[1]
536	218	GEILF4K	4	floating point	Reserved
540	21C	GEILP4K	4	floating point	Reserved
544	220	GEILPFRI	4	floating point	Number of failed 1 MB pageable pages that were requested [1]
548	224	GEILPFCI	4	floating point	Number of demoted 1 MB pageable pages that were converted from 1 MB pages to 4K pages [1]
552	228	GEILCMU	4	floating point	Number of 1 MB high virtual common memory objects whose owner is no longer active[1]
556	22C	GEILCPU	8	floating point	Number of 1 MB high virtual common memory pages whose owner is no longer active[1]
564	234	GEILFPF	4	floating point	Reserved
568	238	*	120	*	Reserved
688	2B0	GEILSMO	8	floating point	Number of memory objects allocated in high virtual shared storage that can be backed in 1 MB frames[1]
696	2B8	GEIRFREM	8	floating point	Number of freemained frames in all address spaces[1]
704	2C0	GEIRGMO	8	floating point	Number of fixed memory objects that are allocated in the system and are backed in 2 GB frames
712	2C8	GEIRGPR	8	floating point	Number of 2 GB pages fixed in central storage
720	2D0	GEIGFUSE	8	floating point	Number of 2 GB frames used by fixed memory objects
728	2D8	GEIGSIZ	8	floating point	Number of 2 GB frames that can be used by fixed 2 GB memory objects
736	2E0	GEILUSE	8	floating point	Number of 1 MB frames in central storage that are in-use by memory objects.
744	2E8	GEILTOT	8	floating point	Total number of 1 MB frames in central storage.
752	2F0	*	8	*	Reserved

Dec Offset	Hex Offset	Name	Length	Format	Description
760	2F8	GEIRUCSASZ	4	binary	IPL size of restricted use common service area (RUCSA) below 16M
764	2FC	GEIERUCSAZ	4	binary	IPL size of RUCSA above 16M
768	300	GEIRUCSAMX	4	binary	Maximum allocated RUCSA below 16M
772	304	GEIERUCSAX	4	binary	Maximum allocated RUCSA above 16M
776	308	GEIRUCSASP	4	binary	Current allocated RUCSA below 16M
780	30C	GEIERUCSAP	4	binary	Current allocated RUCSA above 16M
784	310	GEIRUCSAAV	4	floating point	Accumulated allocated RUCSA below 16M
788	314	GEIERUCSAV	4	floating point	Accumulated allocated RUCSA above 16M
792	318	GEIRUCSARE	4	binary	Amount of unallocated common area left in RUCSA below 16M
796	31C	GEIRUCSAAS	4	floating point	Accumulated allocated RUCSA below 16M (held by the system)[1]
800	320	GEIERUCSAS	4	floating point	Accumulated allocated RUCSA above 16M (held by the system)[1]
804	324	GEIBATRUCSA	4	floating point	Accumulated allocated RUCSA below 16M (held by BATCH initiators)[1]
808	328	GEIBATERUCSA	4	floating point	Accumulated allocated RUCSA above 16M (held by BATCH initiators)[1]
812	32C	GEIASCRUCSA	4	floating point	Accumulated allocated RUCSA below 16M (held by ASCH initiators)[1]
816	330	GEIASCERUCSA	4	floating point	Accumulated allocated RUCSA above 16M (held by ASCH initiators)[1]
820	334	GEIOMVRUCSA	4	floating point	Accumulated allocated RUCSA below 16M (held by OMVS initiators)[1]
824	338	GEIOMVERUCSA	4	floating point	Accumulated allocated RUCSA above 16M (held by OMVS initiators)[1]

Note:

[1] Sum of values obtained at each sample. To obtain average values, divide by the number of valid samples (SSHSMPNR).

ERBIQDG3 - I/O queuing performance data table

Dec offset	Hex offset	Name	Length	Format	Description
0	0	IQDAcr	5	EBCDIC	Acronym 'IQDG3'
5	5	IQDVer	1	binary	IQDG3 version X'01'
6	6	IQDHdrL	2	binary	Length of IQDG3 header
8	8	IQDTotL	4	binary	Total length of IQDG3
12	C	IQDNDat	2	binary	Number of LCU data sections
14	E	IQDLDat	2	binary	Length of LCU data sections
16	10	IQDODat	4	binary	Offset to first LCU data section
20	14	IQDHiLCU	2	binary	Highest LCU number
22	16	IQDNum	2	binary	Number of LCUs

Dec offset	Hex offset	Name	Length	Format	Description
24	18	IQDGFlg	1	binary	Flags **Bit** **Meaning when set** **0** No data **1** ESCON director in configuration **2** Hardware command to retrieve data is not supported **3** DCM supported by hardware **4** Configuration contains DCM managed channel paths **5** Initial command response time supported **6** Extended I/O measurement facility supported **7** Configuration changed
25	19	IQDFLAG	1	binary	IQDG3 flags **Bit** **Meaning when set** **0–7** Reserved
26	1A	*	6	*	Reserved
I/O queuing performance data entry:					
0	0	IQDLCid	2	binary	LCU identification
2	2	IQDLDSt	1	binary	LCU data status **Bit** **Meaning when set** **0** No hardware measurements available. **1** No UCB found for any device of LCU. **2** LCU contains DCM managed channel paths. **3** Path attributes are valid. **4** Extended I/O measurements format. **5 - 7** Reserved.
3	3	IQDDvCl	1	binary	Device class
4	4	IQDLMCMn	2	binary	Minimum number of DCM channel paths used
6	6	IQDLMCMx	2	binary	Maximum number of DCM channel paths used
8	8	IQDLMCDf	2	binary	Defined number of DCM managed channel paths
10	A	IQDLMCUs	2	binary	Currently used number of DCM managed channel paths
Eight channel path data entries. Each entry consists of 16 bytes of configuration information and 28 bytes of measurement data.					
12	C	IQDCPidB	1	binary	Channel path ID

Dec offset	Hex offset	Name	Length	Format	Description

13	D	IQDCPSt	2	binary	Channel path status
					
					Bit
					    Meaning when set
					0
					    Channel path is installed.
					1
					    Channel path is online.
					2
					    Channel path is varied.
					3
					    Path is offline to all devices of this LCU.
					4
					    Vary path during this MINTIME.
					5
					    Channel path is DCM managed.
					6
					    Channel path manipulated, requires reset.
					7
					    Extended I/O measurement block data available.
					8 - 15
					    Reserved.

15	F	IQDCPAt	1	binary	Channel path attribute
16	10	*	2	*	Reserved
18	12	IQDCCUN	2	binary	Number of CUs attached
20	14	IQDCCUA(4)	8	binary	CUs attached to this channel path
28	1C	IQDCCUB	4	binary	CU busy count
32	20	IQDCPT	4	binary	Number of operations accepted on this channel path
36	24	IQDCDPB	4	binary	Number of unsuccessful initial selection attempts due to director port busy
40	28	IQDCCBT	4	binary	CU busy delay time
44	2C	IQDCCMR	4	binary	Initial command response time
48	30	IQDCSBS	4	binary	Switch busy count summation
52	34	*	4	*	Reserved
End of channel path data entries:					
364	16C	IQDLQSM	4	binary	Accumulated CU-Hdr queue length
368	170	IQDLQCT	4	binary	Number of entries placed on the CU-Hdr queue
372	174	IQDLCSt	4	binary	Channel subsystem wait time, in units of 128 microseconds
376	178	IQDLCUB	4	binary	Accumulated CU busy count for DCM managed channel paths
380	17C	IQDLPT	4	binary	Accumulated path taken count for DCM managed channel paths
384	180	IQDLDPB	4	binary	Accumulated director port busy count for DCM managed channel paths
388	184	IQDLCBT	4	binary	Accumulated CU busy delay time for DCM managed channel paths
392	188	IQDLCMR	4	binary	Accumulated initial command response time for DCM managed channel paths
396	18C	IQDLSBS	4	binary	Accumulated switch busy count summation for DCM managed channel paths

Dec offset	Hex offset	Name	Length	Format	Description
400	190	*	4	*	Reserved

ERBLOKG3 - Lock performance data table

Dec offset	Hex offset	Name	Length	Format	Description
0	0	LOKAcr	5	EBCDIC	Acronym 'LOKG3'
5	5	LOKVer	1	binary	LOKG3 version X'01'
6	6	LOKFlag	1	binary	LOKG3 flags **Bit** **Meaning when set** **0–7** Reserved
7	7	*	1	*	Reserved
8	8	LOKHdrL	4	binary	Length of LOKG3 header
12	C	LOKTotL	4	binary	Total length of LOKG3

16	10	LOKLHT	8	binary	Lock holder elements triplet with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      2     Length of one element`
					`  2      2     Number of elements`
					`  4      4     Offset to first element`

24	18	LOKCLT	8	binary	CMS lock elements triplet with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      2     Length of one element`
					`  2      2     Number of elements`
					`  4      4     Offset to first element`

32	20	LOKSHT	8	binary	Spin lock 'held' elements triplet with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      2     Length of one element`
					`  2      2     Number of elements`
					`  4      4     Offset to first element`

40	28	LOKSST	8	binary	Spin lock 'spin' elements triplet with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      2     Length of one element`
					`  2      2     Number of elements`
					`  4      4     Offset to first element`

48	30	LOKLLT	8	binary	Local lock elements triplet with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      2     Length of one element`
					`  2      2     Number of elements`
					`  4      4     Offset to first element`

56	38	*	4	*	Reserved
60	3C	LOKFailC1	2	binary	Cycle gathering failure count for CMS locks
62	4E	LOKFailC2	2	binary	Cycle gathering failure count for local locks
64	40	LOKFailC3	2	binary	Cycle gathering failure count for spin 'held' locks
66	42	LOKFailC4	2	binary	Cycle gathering failure count for spin 'spin' locks

Dec offset	Hex offset	Name	Length	Format	Description
68	44	*	24	*	Reserved
Lock element:					
0	0	Name	8	binary	Name of the lock. In the case of local locks, this is the job name.
8	8	Token	8	EBCDIC	Token of address space in the case of local locks (ASSBSTKN)
16	10	*	8	*	Reserved
24	18	Holder#	4	binary	Number of holders
28	1C	HolderIx	4	binary	Index to first holder element
Holder element:					
0	0	Next	4	binary	Index to next holder element

4	4	HName	10	binary	Name/key of holder element with format:
					
					`Byte  Length  Meaning of triplet field`
					`  0      8     Job name`
					`  8      2     Address space identifier`
					
					For spin lock 'held' elements, the format is:
					
					`Byte  Length  Meaning of triplet field`
					`  0      8     Shared/Exclusive mode`
					`  8      2     CPU identifier`

14	E	*	2	*	Reserved
16	10	ReqAddr	4	binary	Request instruction address
20	14	Held	4	binary	Held counter
24	18	Intr	4	binary	Held + interrupted counter
28	1C	Disp	4	binary	Held + dispatchable counter
32	20	Susp	4	binary	Held + suspended counter

ERBOPDG3 - OMVS process data table

Dec offset	Hex offset	Name	Length	Format	Description
OPDG3 header section:					
0	0	OPDOPDG3	5	EBCDIC	Acronym 'OPDG3'
5	5	OPDVERG3	1	EBCDIC	Control block version X'01'
6	6	OPDHDRLE	2	binary	Length of OPDG3 header
8	8	OPDTOTLE	4	binary	Total length of OPDG3
12	C	OPDENTO	4	binary	Offset to OPDG3 array
16	10	OPDENTL	4	binary	Length of OPDG3 entry
20	14	OPDENTN	4	binary	Number of OPDG3 entries
24	18	OPDSUMO	4	binary	Offset to Summary data (OSDG3)
28	1C	OPDSUML	4	binary	Summary data length

Dec offset	Hex offset	Name	Length	Format	Description

32	20	OPDSTAT	4	binary	Flags:
					
					Bit
					    Meaning When Set
					0
					    OMVS inactive
					1–31
					    Reserved

36	24	*	92	*	Reserved
OSDG3: summary information:					
0	0	OSDPROC	8	EBCDIC	OMVS procedure name
8	8	*	2	*	Reserved
10	A	OSDKASID	2	binary	Kernel address space ID
12	C	OSDPLIST	40	EBCDIC	OMVS parmlib member list
OPDG3 array entry:					
0	0	OPDJOBNM	8	EBCDIC	Job name (as noted in ASCB)
8	8	OPDUSER	8	EBCDIC	User name (from login)
16	10	OPDPID	4	binary	Process ID
20	14	OPDPPID	4	binary	Parent's process ID
24	18	OPDASID	2	binary	Address space ID. Undefined state if 0.
26	1A	*	5	*	Reserved
31	1F	OPDSTYY	4	EBCDIC	4-digit year
35	23	OPDSTDD	3	EBCDIC	3-digit day of year (1-366)
38	26	OPDSTHH	2	EBCDIC	Process start time hour
40	28	OPDSTMM	2	EBCDIC	Process start time minute
42	2A	OPDSTSS	2	EBCDIC	Process start time second
44	2C	OPDCT	8	EBCDIC	Process system and user compute time in STCK format

52	34	OPDSTAT1	1	binary	MVS status flags:
					
					Bit
					    Meaning When Set
					0
					    Space swapped out
					1
					    Ptrace kernel wait
					2–7
					    Reserved

Dec offset	Hex offset	Name	Length	Format	Description

53	35	OPDSTAT2	1	binary	Process status flags:
					
					Bit
					    Meaning When Set
					0
					    Process stopped
					1
					    Reserved
					2
					    multiple threads
					3
					    pthread task in process
					4–7
					    Reserved

54	36	OPDSTAT3	1	EBCDIC	State of reported task:
					
					A
					    Message queue receive wait
					B
					    Message queue sent wait
					C
					    Communication system kernel wait
					D
					    Semaphore operation wait
					E
					    Quiesce frozen
					F
					    File system kernel wait
					G
					    MVS pause wait
					H
					    Multiple threads, pthread_create used
					I
					    Swapped out
					K
					    Other kernel wait
					L
					    Canceled, parent waits
					M
					    Multiple threads, no pthread_create used
					P
					    Ptrace kernel wait
					Q
					    Quiesce termination wait
					R
					    Running
					S
					    Sleeping
					W
					    Waiting for child
					X
					    Creating new process
					Z
					    Zombie. Canceled; parent does not wait

55	37	*	1	*	Reserved
56	38	OPDLWPID	4	binary	Latch process ID the process is waiting for (0 = not waiting)

Dec offset	Hex offset	Name	Length	Format	Description

60	3C	OPDGFLGS	4	binary	General flags:
					Bit
					    Meaning When Set
					0
					    Server information is valid (in fields OPDSNAME, OPDAFILE, OPDMFILE, OPDSTYPE)
					1–31
					    Reserved

64	40	OPDSNAME	32	EBCDIC	Server name in mixed case
96	60	OPDAFILE	4	binary	Number of active files
100	64	OPDMFILE	4	binary	Maximum number of files
104	68	OPDSTYPE	4	binary	Server type
108	6C	OPDCMND	40	EBCDIC	Truncated command buffer in mixed case
148	94	OPDDCT	8	EBCDIC	Delta TCB time
156	9C	OPDDCtIIP	8	Binary	Delta TCB time for zIIP
164	A4	*	4	Binary	Reserved
168	A8	OPDCtIIP	8	Binary	Process system and user compute time on zIIP

ERBPCIG3 - PCIE activity data table

Dec offset	Hex offset	Name	Length	Format	Description
PCIG3 header section:					
0	0	PCIG3Acr	5	EBCDIC	Acronym PCIG3
5	5	PCIG3Ver	1	binary	PCIG3 version
6	6	PCIG3HdL	2	binary	Length of PCIG3 header
8	8	PCIG3Len	4	binary	Total length of PCIG3
12	C	PCIG3Rel	8	EBCDIC	z/OS release level
20	14	PCIG3Num	4	binary	Number of entries
24	18	PCIG3PfL	2	binary	Length of function activity section
26	1A	PCIG3DpfL	2	binary	Length of function type section
28	1C	PCIG3FpgL	2	binary	Length of HW accelerator section
30	1E	PCIG3Fp1L	2	binary	Length of HW accelerator compression section
32	20	PCIG3SYNL	2	binary	Length of synchronous I/O link data section
34	22	PCIG3SRTL	2	binary	Length of synch I/O response time distribution data section
PCIE function activity section:					
0	0	PCIVers	2	binary	Version number
2	2	*	2	*	Reserved
4	4	PCIPfid	4	binary	PCIE function ID (PFID)

Dec offset	Hex offset	Name	Length	Format	Description

8	8	PCIPffl	2	binary	PCIE function status flags
					
					Bit
					    Meaning when set
					0
					    PCIE function is allocated
					1
					    PCIE function is deallocate-pending
					2
					    PCIE function is in permanent error
					3–15
					    Reserved

10	A	PCIPcid	2	binary	Physical/virtual channel id
12	C	PCIDevT	4	EBCDIC	Device type for PCIE function
16	10	PCIDevN	24	EBCDIC	Device name for PCIE function
40	28	PCIJobN	8	EBCDIC	Job name of PCIE function owner
48	30	PCIAsid	2	binary	ASID of PCIE function owner
50	32	PCIPft	1	binary	PCIE function type
51	33	PCIPort	1	binary	ID of the port the PCIE function is attached to
52	34	PCIDmaN	4	binary	Number of DMA address spaces
56	38	PCIATst	8	binary	Timestamp of when PCIE function was allocated
64	40	PCIAllT	4	binary	Amount of time in milliseconds for which the PCIE function was allocated or deallocate-pending
68	44	PCIScnt	4	binary	Sample count for PCI operations
72	48	PCILoop	8	binary	Count of PCI load operations. This value is not reported for synch I/O functions.
80	50	PCIStop	8	binary	Count of PCI store operations. This value is not reported for synch I/O functions.
88	58	PCISbop	8	binary	Count of PCI store block operations. This value is not reported for synch I/O functions.
96	60	PCIRfop	8	binary	Count of PCI refresh translate operations. This value is not reported for synch I/O functions.

104	68	PCIPff1	2	binary	Function status flag 1:
					
					Bit
					    Meaning when set
					0
					    PCIE function is deallocated
					1
					    PCIE function is reallocated
					2–15
					    Reserved

106	6A	*	1	*	Reserved

ERBPCIG3 - PCIE activity data table

Dec offset	Hex offset	Name	Length	Format	Description
107	6B	PCIValidFlag	1	binary	Valid indicators: **Bit** **Meaning when set** **0** PNET IDs are valid **1** PCIE function type is valid **2** Operation rates are invalid **3** Global performance reporting mode is enabled **4–7** Reserved
108	6C	*	16	*	Reserved
124	7C	PCIUtilityStrings	32	EBCDIC	Utility strings
156	9C	*	8	*	Reserved
PCIE function type section:					
0	0	PCIDBYR	8	binary	Function specific count: **Fmt** **Meaning** **1** Number of bytes received **2** Number of work units processed **3** Unused **4** Number of bytes read **0** DMA read counter
8	8	PCIDBYT	8	binary	Function specific count: **Fmt** **Meaning** **1** Number of bytes transmitted **2** Max. number of work units that the PCIE function can process **3** Number of bytes transmitted **4** Number of bytes written **0** DMA write counter

Dec offset	Hex offset	Name	Length	Format	Description

16	10	PCIDFMT	1	binary	Format of this entry:
					
					Fmt
					    Meaning
					0
					    RoCE or zEDC on zEC12/zBC12
					1
					    RoCE on post Fmt-0 hardware
					2
					    zEDC on post Fmt-0 hardware
					3
					    ISM on post Fmt-0 hardware
					4
					    Synchronous I/O on post Fmt-0 hardware

17	11	*	7	*	Reserved

24	18	PCIDPKR	8	binary	Function specific count:
					
					Fmt
					    Meaning
					1
					    Number of packets received
					4
					    Number of successful requests
					0, 2, 3
					    Unused

32	20	PCIDPKT	8	binary	Function specific count:
					
					Fmt
					    Meaning
					1
					    Number of packets transmitted
					4
					    Number of local rejects
					0, 2, 3
					    Unused

40	28	PCISRRF	8	binary	Function specific count:
					
					Fmt
					    Meaning
					4
					    Number of remote rejects
					0, 1, 2, 3
					    Unused

48	30	PCISTPF	8	binary	Function specific count:
					
					Fmt
					    Meaning
					4
					    Total processing time in microseconds
					0, 1, 2, 3
					    Unused

Dec offset	Hex offset	Name	Length	Format	Description

56	38	PCISRBC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Number of bytes read by all synchronous I/O functions that are using this synchronous I/O link on this CPC.
					
					0, 1, 2, 3
					    Unused

64	40	PCISWBC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Number of bytes written by all synchronous I/O functions that are using this synchronous I/O link on this CPC.
					
					0, 1, 2, 3
					    Unused

72	48	PCISSRC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Number of requests successfully processed by all synch I/O functions using this synchronous I/O link function on this CPC.
					
					0, 1, 2, 3
					    Unused

80	50	PCISLRC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Number of local rejects of all synchronous I/O functions that are using this synchronous I/O link on this CPC.
					
					0, 1, 2, 3
					    Unused

88	58	PCISRRC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Number of remote rejects of all synchronous I/O functions that are using this synchronous I/O link on this CPC.
					
					0, 1, 2, 3
					    Unused

Dec offset	Hex offset	Name	Length	Format	Description

96	60	PCISTPC	8	binary	Function specific count. Only valid in global performance reporting mode:
					
					Fmt
					    Meaning
					
					4
					    Total processing time in microseconds of all synchronous I/O functions are using this synchronous I/O link on this CPC.
					
					0, 1, 2, 3
					    Unused

Hardware accelerator section:					
0	0	PCIFTyp	4	binary	HW accelerator application type
4	4	PCIFDsc	32	EBCDIC	HW accelerator application description
36	24	*	12	*	Reserved
48	30	PCIFRqc	4	binary	Total number of requests that completed successfully
52	34	PCIFRqe	4	binary	Total number of requests that completed with an error
56	38	PCIFQfl	4	binary	Number of times that the adapter queue was full when a new request was submitted
60	3C	*	4	*	Reserved
64	40	PCIFTet	8	binary	Total execution time of all requests in microseconds
72	48	PCIFSqe	16	binary	Sum of the squares of the individual execution times
88	58	PCIFTqt	8	binary	Total queue time of all requests in microseconds
96	60	PCIFSqq	16	binary	Sum of the squares of the individual queue times
112	70	PCIFDrd	8	binary	Total DMA reads in units of 256 bytes
120	78	PCIFDwr	8	binary	Total DMA writes in units of 256 bytes
Hardware accelerator compression section:					
0	0	PCI1Dib	8	binary	Total number of deflate input bytes
8	8	PCI1Dis	16	binary	Sum of the squares of the individual deflate input bytes
24	18	PCI1Dob	8	binary	Total number of deflate output bytes
32	20	PCI1Dos	16	binary	Sum of the squares of the individual deflate output bytes
48	30	PCI1Iib	8	binary	Total number of inflate input bytes
56	38	PCI1Iis	16	binary	Sum of the squares of the individual inflate input bytes
72	48	PCI1Iob	8	binary	Total number of inflate output bytes
80	50	PCI1Ios	16	binary	Sum of the squares of the individual inflate output bytes
96	60	PCI1Dct	4	binary	Total number of deflate requests
100	64	PCI1Ict	4	binary	Total number of inflate requests
104	68	PCI1Bpc	8	binary	Cumulative size of memory in bytes for in-use buffers at the time of each request
112	70	PCI1Bps	4	binary	Total size of memory in bytes allocated to the buffer pool
Synchronous I/O link data section:					
0	0	PCISynND	26	EBCDIC	Node descriptor of the storage controller the synchronous I/O link is connected to
Synchronous I/O response time distribution data section:					
0	0	PCIRTDReadCnt	2	binary	Number of response time distribution read buckets

Dec offset	Hex offset	Name	Length	Format	Description
2	2	PCIRTDWriteCnt	2	binary	Number of response time distribution write buckets
4	4	PCIRTDBckt (30)	8	binary	Read and write response time distribution buckets
4	4	PCIRTDRngVal	4	binary	Response time distribution bucket range value
8	8	PCIRTDSmpCnt	4	binary	Response time distribution bucket sample count

ERBRCDG3 - Resource collection data

Resource collection data header

Dec Offset	Hex Offset	Name	Length	Format	Description
0	0	RCDACRO	5	EBCDIC	Acronym 'RCDG3'
5	5	RCDVERS	1	binary	RCDG3 version
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.

Dec Offset	Hex Offset	Name	Length	Format	Description
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	RCDMODE	1	binary	Mode flags
					
					Bit
					    Meaning when set
					0
					    zAAP honor priority
					1
					    zIIP honor priority
					2
					    RCDG3 converted to lower service level
					3
					    RCDG3 converted to higher release or service level
					4–7
					    Reserved

95	5F	RCDSUPP	1	binary	Data availability flags
					
					Bit
					    Meaning when set
					0
					    Page residency time
					1–7
					    Reserved

96	60	RCDMADJ	4	binary	Value of RMCTADJC - adjustment factor for CPU rate
100	64	RCDNFFI	4	binary	Normalization factor for zAAP. Multiply zAAP times or service units with this value and divide by 256 to calculate the CP equivalent value.
104	68	RCDNFFS	4	binary	Normalization factor for zIIP. Multiply zIIP service units with this value and divide by 256 to calculate the CP equivalent value.
108	6C	RCDPADJSCF	4	binary	Scaling factor for RCDPADJ.
112	70	RCDPADJ	4	binary	Physical CPU adjustment factor for CP processors.
116	74	RCDPADJCBP	4	binary	Reserved.

Response time distribution map array

Dec Offset	Hex Offset	Name	Length	Format	Description
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

Dec Offset	Hex Offset	Name	Length	Format	Description
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.
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

Dec Offset	Hex Offset	Name	Length	Format	Description
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. **3–7** 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.
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)

Dec Offset	Hex Offset	Name	Length	Format	Description
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)
100	64	RCDRCT	4	binary	Total region control task time in microsecond units
104	68	RCDIIT	4	binary	Total I/O interrupt time in microsecond units
108	6C	RCDHST	4	binary	Total hiperspace service time in microsecond units
112	70	RCDIFAT	8	binary	Reserved
120	78	RCDIFCT	8	binary	Reserved
128	80	RCDIFASU	8	binary	Total zAAP service units. Multiply with RCDNFFI and divide by 256 to calculate the CP equivalent value
136	88	RCDIFASUCP	8	binary	Total zAAP service units spent on CPs
144	90	RCDSUPSU	8	binary	Total zIIP service units. Multiply with RCDNFFS and divide by 256 to calculate the CP equivalent value
152	98	RCDSUPSUCP	8	binary	Total zIIP service units spent on CPs
160	A0	RCDTPDP	8	binary	Total CPU time spent for work units with promoted dispatching priority (in 1024-microsecond units).
168	A8	RCDCPUDL	4	binary	CP delay samples
172	AC	RCDAAPDL	4	binary	zAAP delay samples
176	B0	RCDIIPDL	4	binary	zIIP delay samples
180	B4	RCDRGCAP	4	binary	Resource group capping delay samples
184	B8	*	52	*	Reserved
236	EC	RCDTETX	8	binary	Total transaction elapsed time (in microseconds)
244	F4	RCDXETX	8	binary	Total transaction execution time (in microseconds)
252	FC	RCDQDTX	8	binary	Queue delay time (in microseconds)
260	104	RCDADTX	8	binary	Resource affinity delay time (in microseconds)
268	10C	RCDCVTX	8	binary	JCL conversion delay time (in microseconds)
276	114	RCDIQTX	8	binary	Ineligible queue time (in microseconds)
284	11C	RCDRTDM	4	binary	Midpoint of response time distribution. Equal to goal if period with response time goal (unit as indicated in SVPG3). Zero if discretionary or system goal or no goal defined.
288	120	RCDPRS	8	floating point	Page residency time (in 1024 microsecond units)
296	128	RCDCIOU	4	binary	Total I/O usings. These are included in RCDCUSE. Only non-paging DASD I/O can contribute to I/O usings.
300	12C	RCDCIOD	4	binary	DASD I/O delay samples
304	130	RCDCIDL	4	binary	Idle samples. Work is in STIMER wait, TSO terminal wait, APPC wait, or is an initiator waiting for work. These samples are not included in RCDTOTD.
308	134	RCDCUNK	4	binary	Unknown samples. Dispatchable unit or address space is waiting, but none of the delays (listed earlier) apply.

Dec Offset	Hex Offset	Name	Length	Format	Description
312	138	RCDENCTRXNUM	4	binary	Number of subsystem transactions processed within enclaves.
316	13C	RCDENCTRXCALLS	4	binary	Number of times transaction data has been reported by subsystem work managers when deleting an enclave. When zero, no transaction data for enclaves has been provided by the subsystem work manager.
320	140	RCDENCTRXET	8	binary	Total execution time, in microseconds, for all subsystem transactions reported in RCDENCTRXNUM.
328	148	RCDENCTRXETS	8	binary	Sum of squared execution times, in microseconds, for all subsystem transactions reported in RCDENCTRXNUM.

Resource collection data - response time distribution array

Dec Offset	Hex Offset	Name	Length	Format	Description
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 versus goals specified.

Resource collection data - subsystem work manager delays

Dec Offset	Hex Offset	Name	Length	Format	Description
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
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 available to the work manager that is allowed to run.
24	18	RCDWLOK	4	binary	Total number of waiting for lock state samples

Dec Offset	Hex Offset	Name	Length	Format	Description
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
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	RCDBPMI	4	binary	Number of state samples representing Db2 buffer pool misses that resulted in I/O.
80	50	*	12	*	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

Dec Offset	Hex Offset	Name	Length	Format	Description
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	RCDWTY6	4	binary	Total number of waiting for resource type 6 samples
136	88	RCDWTY7	4	binary	Total number of waiting for resource type 7 samples
140	8C	RCDWTY8	4	binary	Total number of waiting for resource type 8 samples
144	90	RCDWTY9	4	binary	Total number of waiting for resource type 9 samples
148	94	RCDWT10	4	binary	Total number of waiting for resource type 10 samples
152	98	RCDWT11	4	binary	Total number of waiting for resource type 11 samples
156	9C	RCDWT12	4	binary	Total number of waiting for resource type 12 samples
160	A0	RCDWT13	4	binary	Total number of waiting for resource type 13 samples
164	A4	RCDWT14	4	binary	Total number of waiting for resource type 14 samples
168	A8	RCDWT15	4	binary	Total number of waiting for resource type 15 samples

ERBREDG3 - Resource data record

Dec offset	Hex offset	Name	Length	Format	Description

0	0	REDENTRY(10)	12	EBCDIC	RED Array
					
					Entry
					    Resource
					1
					    USER
					2
					    PROCESSOR
					3
					    DEVICE
					4
					    STORAGE
					5
					    JES2/JES3
					6
					    HSM
					7
					    ENQ
					8
					    MOUNT
					9
					    MESSAGE
					10
					    XCF

REDENTRY section:					
0	0	REDREDID	1	binary	Resource Data Record ID

ERBSCMG3 - Storage class memory data table

Dec offset	Hex offset	Name	Length	Format	Description
1	1	REDFLAG1	1	binary	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	REDUWDLE	2	binary	For all resources except ENQ: Length of USE/WAI 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

ERBSCMG3 - Extended Asynchronous Data Mover (EADM) data table

Dec offset	Hex offset	Name	Length	Format	Description
SCMG3 header section:					
0	0	SCM_EyeC	5	EBCDIC	Acronym SCMG3
5	5	SCM_Ver	1	binary	SCMG3 version
6	6	*	2	*	Reserved
8	8	SCM_HdrL	4	binary	Length of SCMG3 header
12	C	SCM_TotL	4	binary	Total length of SCMG3
16	10	*	2	*	Reserved
18	12	SCM_DIL	2	binary	Length of SCM DI entry
20	14	SCM_DIO	4	binary	Offset to SCM DI entry
24	18	SCM_CML	2	binary	Length of SCM CM entry
26	1A	SCM_CMN	2	binary	Number of SCM CM entries
28	1C	SCM_CMO	4	binary	Offset to SCM CM entries
EADM device information entry (SCM DI):					
0	0	SCMDI_EyeC	5	EBCDIC	Eyecatcher SCMDI
5	5	SCMDI_Ver	1	binary	SCMDI version
6	6	*	2	*	Reserved

Dec offset	Hex offset	Name	Length	Format	Description
8	8	SCMDI_Body	80	binary	EADM device information data
					For information about all fields contained in the SCMDI_Body section, see SMF record type 74 subtype 10, Extended asynchronous data mover (EADM) device (subchannel) information in *z/OS MVS System Management Facilities (SMF)*.
					Note that SCMDI_Body is at offset 8 in the EADM Device Information Entry section of SCMG3. This offset must be added to offsets of R7410DI when accessing fields of SCMDI in SCMG3.
8	8	R7410DSCT	4	binary	SSCH count across all devices
12	C	R7410DNUM	4	binary	Number of updates to the time accumulation fields
16	10	R7410DFPT	8	binary	Sum of function pending times across all devices in units of 128 microseconds (doubleword format).
24	18	R7410DIQT	8	binary	Sum of IOP queue times across all devices in units of 128 microseconds (doubleword format).
32	20	R7410DCRT	8	binary	Sum of initial command response times across all devices in units of 128 microseconds (doubleword format).
SCM configuration measurement entry (SCM CM):					
0	0	SCMCM_EyeC	5	EBCDIC	Eyecatcher SCMCM
5	5	SCMCM_Ver	1	binary	SCMCM version
6	6	*	2	*	Reserved
8	8	SCMCM_Body	56	binary	SCM configuration measurement data
					For information about all fields contained in the SCMCM_Body section, see SMF record type 74 subtype 10, Storage class memory (SCM) configuration measurement section in *z/OS MVS System Management Facilities (SMF)*.
					Note that SCMCM_Body is at offset 8 in the SCM Configuration Measurement Entry section of SCMG3. This offset must be added to offsets of R7410CM when accessing fields of SCMCM in SCMG3.
8	8	R7410CRID	2	binary	SCM resource identifier
10	A	R7410CPID	2	binary	Part identifier
12	C	R7410CDUS	4	binary	Data unit size in bytes
16	10	R7410CRQC	4	binary	Internal requests processed at CPC level
20	14	R7410CRQ	4	binary	Internal requests processed at LPAR level
24	18	R7410CDWC	4	binary	Data units written at CPC level
28	1C	R7410CDW	4	binary	Data units written at LPAR level
32	20	R7410CDRC	4	binary	Data units read at CPC level
36	24	R7410CDR	4	binary	Data units read at LPAR level
40	28	R7410CRTC	4	binary	Aggregate time spent on execution of requests involving resource part in units of 128 microseconds at CPC level
44	2C	R7410CRT	4	binary	Aggregate time spent on execution of requests involving resource part in units of 128 microseconds at LPAR level
48	30	R7410CIQC	4	binary	Accumulated IOP queue time in units of 128 microseconds at CPC level
52	34	R7410CWUC	4	binary	Utilization at CPC level.
56	38	R7410CWU	4	binary	Utilization at LPAR level.

Dec offset	Hex offset	Name	Length	Format	Description

60	3C	R7410FLG	1	binary	Flag byte:
					
					Bit
					    Meaning when set
					0
					    SCM resource type is VFM
					1–7
					    Reserved

61	3D	*	3	*	Reserved

ERBSHDG3 - Sample header

Dec Offset	Hex Offset	Name	Length	Format	Description
0	0	SHDSHDG3	5	EBCDIC	Acronym 'SHDG3'
5	5	SHDRMFV	1	binary	Control block version X'03'
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

ERBSPGG3 - Storage group and volume data

Dec offset	Hex offset	Name	Length	Format	Description
SPGG3 header section:					
0	0	SPGACR	5	EBCDIC	Acronym 'SPGG3'
5	5	SPGVER	1	binary	Control block version X'01'
6	6	*	2	*	Reserved
8	8	SPGHDRL	4	binary	Length of SPGG3 header
12	C	SPGTOTL	4	binary	Total length of SPGG3

Dec offset	Hex offset	Name	Length	Format	Description
16	10	SPGSGDATL	4	binary	Length of one storage group entry
20	14	SPGSGDATN	4	binary	Number of storage group entries
24	18	SPGSGDATO	4	binary	Offset to storage group entries
28	1C	SPGVOLDATL	4	binary	Length of one volume data entry
32	20	SPGVOLDATN	4	binary	Number of volume data entries
36	24	SPGVOLDATO	4	binary	Offset to volume data entries
40	28	SPGSTAT	2	binary	Status flags **Bit** **Meaning when set** **0** No SPG data collected **1** Internal problem **2** SMS inactive **3–7** Reserved **8** No volume data available **9** no storage group data **10–15** Reserved
42	2A	*	6	*	Reserved
Storage group entry:					
0	0	GNAMEL	2	binary	Actual length of storage group name
2	2	GNAME	30	EBCDIC	Storage group name
32	20	FIRSTVOL	2	binary	Index of first volume entry for this storage group
34	22	NUMBERVOL	2	binary	Number of volume entries for this storage group
36	24	*	4	*	Reserved
Volume data entry:					
0	0	VNAMEL	2	binary	Actual length of volume name
2	2	VNAME	6	EBCDIC	Volume name (VOLSER)
8	8	TOTALSPACE	4	binary	Total space on volume (megabyte)
12	C	FREESPACE	4	binary	Free space on volume (megabyte)
16	10	LBLOCKSIZE	4	binary	Largest block of unallocated space (megabyte)
20	14	*	4	*	Reserved

ERBSSHG3 - MINTIME set of samples header

Dec offset	Hex offset	Name	Length	Format	Description
SSHG3 header section:					
0	0	SSHSSHG3	5	EBCDIC	Acronym 'SSHG3'
5	5	SSHRMFV	1	binary	Control block version X'0E'
6	6	SSHLEN	2	binary	Length of SSHG3 header

ERBSSHG3 - Samples header

Dec offset	Hex offset	Name	Length	Format	Description
8	8	SSHRMFVN	3	EBCDIC	Monitor III data gatherer version number

11	B	SSHFLAG1	1	binary	Flag byte
					Bit
					    Meaning
					0
					    Data are compressed
					1
					    WLM 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 to ASID table from ERBSSHG3
96	60	*	12	*	Reserved
108	6C	SSHDVTO	4	binary	Offset to DVT table from ERBSSHG3
112	70	*	8	*	Reserved
120	78	SSHENTO	4	binary	Offset to ENT table from ERBSSHG3
124	7C	*	24	*	Reserved
148	94	SSHGEIO	4	binary	Offset to GEIG3 table from ERBSSHG3

152	98	SSHIOML	1	binary	Processor type on which data was created
					Value
					    Meaning
					X'03'
					    9672, zSeries

153	99	SSHEFLAG	1	binary	Extended storage indicators
					Bit
					    Meaning When Set
					0
					    Extended storage installed
					1–7
					    Reserved

Dec offset	Hex offset	Name	Length	Format	Description

154	9A	SSHPRFGS	1	binary	Processor flags
					
					Bit
					    Meaning When Set
					0
					    ES/Connection Channel enabled
					1
					    ES/Connection Director configured
					2–7
					    Reserved

155	9B	*	1	*	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	Job name of HSM subsystem
248	F8	SSHHSMAS	2	binary	ASID number of HSM subsystem
250	FA	SSHJESJN	8	EBCDIC	Job name of JES subsystem
258	102	SSHJESAS	2	binary	ASID number of JES subsystem
260	104	*	8	*	Reserved
268	10C	SSHCSRO	4	binary	Offset to CSR table from ERBSSHG3. 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 table from ERBSSHG3
284	11C	SSHCPUO	4	binary	Offset to CPUG3 table from ERBSSHG3
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 table from ERBSSHG3
308	134	*	8	*	Reserved
316	13C	SSHCFIO	4	binary	Offset to CFIG3 table from ERBSSHG3
320	140	SSHCATO	4	binary	Offset to CATG3 table from ERBSSHG3

ERBSVPG3 - Service policy

Dec offset	Hex offset	Name	Length	Format	Description
324	144	SSHVRIO	4	binary	Offset to VRIG3 table from ERBSSHG3
328	148	SSHOPDO	4	binary	Offset to OPDG3 table from ERBSSHG3
332	14C	*	4	*	Reserved
336	150	SSHSPGO	4	binary	Offset to SPGG3 table from ERBSSHG3
340	154	SSHCPDO	4	binary	Offset to CPDG3 table from ERBSSHG3
344	158	SSHIQDO	4	binary	Offset to IQDG3 table from ERBSSHG3
348	15C	SSHXCFO	4	binary	Offset to XCFG3 table from ERBSSHG3
352	160	SSHLOKO	4	binary	Offset to LOKG3 table from ERBSSHG3
356	164	SSHPCIO	4	binary	Offset to PCIG3 table from ERBSSHG3
360	168	SSHSCMO	4	binary	Offset to SCMG3 table from ERBSSHG3
364	16C	SSHZFXO	4	binary	Offset to ZFXG3 table from ERBSSHG3
368	170	SSHCRYO	4	binary	Offset to CRYG3 table from ERBSSHG3
372	174	*	12	*	Reserved

ERBSVPG3 - Service policy

Dec offsets	Hex offsets	Name	Length	Format	Description
SVPG3 header section:					
0	0	SVPNAM	5	EBCDIC	Acronym 'SVPG3'
5	5	SVPDVN	1	binary	Control block version
6	6	SVPDIL	2	binary	Length of SVPG3 header
8	8	SVPDLE	4	binary	Total length of the active service policy data structure
12	C	SVPTIB	8	EBCDIC	Begin time in TOD. Time of policy activation
20	14	SVPTIE	8	EBCDIC	End time in TOD. Time of policy deactivation
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

Dec offsets	Hex offsets	Name	Length	Format	Description
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	User ID of the system operator or service administrator who activated the service policy
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	User ID 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

Dec offsets	Hex offsets	Name	Length	Format	Description
48	30	SVPCRN	8	EBCDIC	Name of the resource group this service class is associated with - blanks if no resource group association
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	SVPCFL	2	binary	Class flags **Bit** **Meaning when set** **0** Reserved **1** Indicator for CPU critical **2** Indicator for Storage Protection **3-7** Reserved **8** Specialty engine eligible work in this service class will not be offloaded to CPs for help processing **9–15** 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.

Dec offsets	Hex offsets	Name	Length	Format	Description
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 bit 1 of SVPGLT is ON, this field contains the minimum capacity of the resource group in unweighted CPU service units per second. In addition, the scope of the resource group is sysplex-wide. See also the description of bit 3, bit 4, and bit 7 of SVPGLT.
44	2C	SVPGMX	4	binary	If bit 0 of SVPGLT is ON, this field contains the maximum capacity of the resource group in unweighted CPU service units per second. In addition, the scope of the resource group is sysplex-wide. See also the description of bit 3, bit 4, and bit 7 of SVPGLT.
48	30	SVPGLT	4	binary	Indicators **Bit** 　　**Meaning when set** **0** 　　Maximum capacity was specified **1** 　　Minimum capacity was specified **2** 　　Reserved **3** 　　Specification of SVPGMN and SVPGMX is in % of the LPAR share. The scope of the resource group is system-wide rather than sysplex-wide. **4** 　　Specification of SVPGMN and SVPGMX is in % of a single processor (CP) capacity. The scope of the resource group is system-wide rather than sysplex-wide. **5** 　　Memory limit was specified **6** 　　Specialty processor consumption is included into the WLM capping algorithms, i.e. SVPGMN and SVPGMX limit the combined general purpose and specialty processor consumption. **7** 　　Specification of SVPGMN and SVPGMX is in MSU/h. **8-31** 　　Reserved
52	34	SVPMLIM	4	binary	Maximum memory limit in GB
56	38	SVPGL1	4	binary	Resource group level indicators **Bit** 　　**Meaning when set** **0** 　　Indicator for a tenant resource group **1–31** 　　Reserved
60	3C	*	4	*	Reserved
64	40	SVPGTI	8	EBCDIC	Tenant identifier. Only valid if bit 0 of SVPGL1 is ON.
72	48	SVPGTN	32	EBCDIC	Tenant name. Only valid if bit 0 of SVPGL1 is ON.

Dec offsets	Hex offsets	Name	Length	Format	Description
104	68	SVPGSID	64	EBCDIC	Solution ID. Only valid if bit 0 of SVPGL1 is ON.
Report class information:					
0	0	SVPRNM	8	EBCDIC	Report class name
8	8	SVPRDE	32	EBCDIC	Report class description

40	28	SVPRFL	4	binary	Indicators
					
					Bit
					    Meaning when set
					0
					    Indicator for a tenant report class
					1–31
					    Reserved

44	2C	SVPRGI	4	binary	Resource group index – the index of the tenant resource group entry in SVPRG of the resource group to which this tenant report class belongs.
48	30	SVPRGN	8	EBCDIC	Name of the tenant resource group that is associated with this tenant report class. Blanks if no tenant resource group is associated.

ERBUWDG3 - USE/WAIT record

Offsets Dec	Hex	Name	Length	Format	Description

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
					    Processor was used by enclaves
					1
					    Processor was a zAAP
					2
					    Processor was a standard processor used by zAAP work
					3
					    Processor was a standard processor
					4
					    Processor was a zIIP
					5
					    Processor was a standard processor used by zIIP work
					6-7
					    Reserved

Extended data for DEV section (see resource id in UWDUWRID):					

Offsets Dec	Hex	Name	Length	Format	Description
3	3	UWDDEVNR	2	binary	Reserved
5	5	*	3	binary	Reserved
8	8	UWDDEVN4	4	binary	Device table (DVTG3) index
Extended data for STOR section (see resource id in UWDUWRID):					
3	3	UWDPDEVR	2	binary	Reserved
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
6	6	UWDPDEV4	4	binary	Paging Device table (DVTG3) index
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 Resource Measurement Facility 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 Resource Measurement Facility 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 Dec	Hex	Name	Length	Format	Description

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	
5	5	UWDDEVI4	4	binary	Mount Device table (DVTG3) 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

ERBVRIG3 - VSAM RLS information data table

Dec offset	Hex offset	Name	Length	Format	Description
0	0	VRIVRIG3	5	EBCDIC	Acronym 'VRIG3'
5	5	VRIVERG3	1	binary	VRIG3 version X'04'
6	6	VRIHDRLE	2	binary	Length of VRIG3 header
8	8	VRITOTLE	4	binary	Total VRIG3 length
12	C	VRISYSNA	8	binary	z/OS system name
20	14	VRISCEOF	4	binary	Offset to first entry for SMF type 42 subtype 15
24	18	VRISCELE	2	binary	Length of one entry for SMF type 42 subtype 15
26	1A	VRISCENR	2	binary	Number of entries for SMF type 42 subtype 15
28	1C	VRIVSEOF	4	binary	Offset to first entry for SMF type 42 subtype 16
32	20	VRIVSELE	2	binary	Length of one entry for SMF type 42 subtype 16
34	22	VRIVSENR	2	binary	Number of entries for SMF type 42 subtype 16

Dec offset	Hex offset	Name	Length	Format	Description
36	24	VRILREOF	4	binary	Offset of first local buffer LRU section entry
40	28	VRILRELE	2	binary	Length of one LRU system entry
42	2A	VRILRENR	2	binary	Number of LRU system entries
44	2C	*	4	*	Reserved

48	30	VRIFLAG	1	binary	Flags
					
					Bit
					    Meaning when set
					0
					    SMF type 42 subtype 16 data requested.
					1
					    Data sections above the bar are present.
					2
					    Multi lock structures are supported.
					3–7
					    Reserved

49	31	*	23	*	Reserved
Storage class related data from SMF type 42 subtypes 15 and 16:					
0	0	VRIENSTC	8	binary	Storage class name
8	8	VRIENCSN	8	binary	Cache set name
16	10	VRIENCCN	16	binary	DFP cache structure name
32	20	VRILELTO	4	binary	Number of record lock requests
36	24	VRILELTC	4	binary	Number of record lock requests that caused true contention
40	28	VRILELFC	4	binary	Number of record lock requests that caused false contention
44	2C	VRIDEBWR	4	binary	Number of direct access BMF write requests
48	30	VRIDEBRH	4	binary	Number of direct access BMF read hits
52	34	VRIDEBRV	4	binary	Number of direct access BMF valid read hits
56	38	VRIDEBFI	4	binary	Number of direct access BMF false invalids
60	3C	VRIDECRH	4	binary	Number of direct access CF read hits
64	40	VRIDERRD	4	binary	Number of READ real I/O direct requests to DASD
68	44	VRIDEREQ	4	binary	Number of direct requests
72	48	VRIDETOT	8	binary	Total amount of time for all direct access requests in this interval
80	50	VRISEBWR	4	binary	Number of sequential access BMF write requests
84	54	VRISEBRH	4	binary	Number of sequential access BMF read hits
88	58	VRISEBRV	4	binary	Number of sequential access BMF valid read hits
92	5C	VRISEBFI	4	binary	Number of sequential access BMF false invalids
96	60	VRISECRH	4	binary	Number of sequential access CF read hits
100	64	VRISERRD	4	binary	Number of READ real I/O sequential requests to DASD
104	68	VRISEREQ	4	binary	Number of sequential requests
108	6C	VRISETOT	8	binary	Total amount of time for all sequential access requests in this interval

Dec offset	Hex offset	Name	Length	Format	Description

116	74	VRIENFLG	1	binary	Flags
					
					Bit
					    Meaning when set
					0
					    Access to buffers above the bar
					1– 7
					    Reserved

117	75	*	3	*	Reserved
120	78	VRIENLCN	16	binary	Lock structure name
136	88	VRIENLSN	8	binary	Lock set name
Data set data from SMF type 42 subtype 16:					
0	0	VRIVSVSN	44	EBCDIC	VSAM sphere name
44	2C	VRIVSDSN	44	EBCDIC	Data set name
88	58	VRIVSRC	4	binary	Number of REDOs
92	5C	VRIVSRRC	4	binary	Number of recursive REDOs
96	60	VRIVSWBC	4	binary	Number of BMF writes
100	64	VRIVSSRR	4	binary	Number of SCM read requests
104	68	VRIVSSRC	4	binary	Number of SCM read requests that encountered castout lock contention

108	6C	VRIVSFLG	1	binary	Status flags
					
					Bit
					    Meaning when set
					0
					    Accesses buffers above the bar
					1–7
					    Reserved

109	6D	*	3	*	Reserved
Local buffer LRU section entry:					
0	0	VRILRTCT	8	binary	Total CPU time for this record, in milliseconds
8	8	VRILRBSG	4	binary	Buffer size goal current goal, in MB. Format is short floating point if bit 1 of VRILRFLG is set.
12	C	VRILRBSH	4	binary	Buffer size goal actual goal, in MB. Format is short floating point if bit 1 of VRILRFLG is set.
16	10	VRILRBOC	4	binary	Number of buffer manager LRU intervals where BMF was over the goal (critical)
20	14	VRILRBOG	4	binary	Number of buffer manager LRU intervals where BMF was over the goal and normal algorithms were bypassed to reclaim buffers
24	18	VRILRBIN	4	binary	Number of buffer manager LRU intervals processed
28	1C	VRILRBTN	4	binary	Total number of times that BMF was called on this interval
32	20	VRILRBHC	4	binary	Buffer manager number of 'hits' during this interval
36	24	VRILRSHC	4	binary	Sysplex cache manager number of 'hits' during this interval
40	28	VRILRDHC	4	binary	DASD number of 'hits' during this interval
44	2C	VRILRSRC	4	binary	Number of SCM read requests during this interval
48	30	VRILRSCR	4	binary	Number of SCM read requests which encountered castout
52	34	VRILRRC	4	binary	Number of REDOs during this interval

Dec offset	Hex offset	Name	Length	Format	Description
56	38	VRILRRRC	4	binary	Number of recursive REDOs during this interval
60	3C	VRILRWTN	4	binary	Total number of write requests
64	40	VRILPBUF	192	binary	Buffer pool array consisting of 16 entries. Each entry is 12 bytes long. The first entry represents the 2k storage pool. The second entry is for the 4k storage pool. The next entries represent the 6k, 8k, 10k, 12k, 14k, 16k, 18k, 20k, 22k, 24k, 26k, 28k, 30k, and the last entry is for the 32k storage pool._x000D_Three fields are reported in each entry:_x000D_**Bytes Field name Description**_x000D_0-3 VRILPLBC Low value of the number of BMF buffers for this pool_x000D_4-7 VRILPHBC High value of the number of BMF buffers for this pool_x000D_8-11 VRILPCBC Current value of the number of BMF buffers for this pool
256	100	VRILRFPL	4	binary	Minimum total number of fixed pages. Format is short floating point if bit 1 of VRILRFLG is set.
260	104	VRILRFPH	4	binary	Maximum total number of fixed pages. Format is short floating point if bit 1 of VRILRFLG is set.
264	108	VRILRFPA	4	binary	Average total number of fixed pages. Format is short floating point if bit 1 of VRILRFLG is set.
268	10C	VRILRFST	4	binary	Fixed storage amount, in MB. Format is short floating point if bit 1 of VRILRFLG is set.
272	110	VRILRRST	4	binary	Real storage percentage
273	111	VRILRFLG	1	binary	Status flags_x000D_**Bit**_x000D_**Meaning when set**_x000D_**0**_x000D_Accesses buffers above the bar_x000D_**1**_x000D_Buffer size, in short floating point format_x000D_**2-7**_x000D_Reserved
274	112	*	3	*	Reserved

ERBXCFG3 - XCF Activity data table

Offsets Dec	Hex	Name	Length	Format	Description
XCFG3 header section:					
0	0	XCFAcr	5	EBCDIC	Acronym XCFG3
5	5	XCFVER	1	binary	XCFG3 version
6	6	XCFHdrL	2	binary	Length of XCFG3 header
8	8	XCFTotL	4	binary	Total length of XCFG3
12	C	XCFSYSN	8	EBCDIC	System name
20	14	XCFSID	4	EBCDIC	SMF system ID
24	18	XCFPART	8	EBCDIC	Partition name

ERBXCFG3 - XCF activity data table

Offsets Dec	Hex	Name	Length	Format	Description
32	20	XCFREL	8	EBCDIC	z/OS release level
40	28	XCFINTM	4	binary	XCF Monitoring interval
44	2C	XCFINTO	4	binary	XCF Operator interval
48	30	XCFSTAT	1	binary	XCF System Status **Bit** Meaning When Set **0** Reserved **1** Active **2** Status-update missing **3** In Sysplex partitioning **4** Single system **5** In cleanup processing **6–7** Reserved
49	31	XCFESTAT	1	binary	Extended status flag **Bit** Meaning When Set **0** No IXCQUERY data **1** Data Server master system **2–7** Reserved
50	32	*	66	*	Reserved
116	74	XCFGDatL	2	binary	Length of group data entry
118	76	XCFGDatN	2	binary	Number of group data entries
120	78	XCFGDatO	4	binary	Offset to first group data entry
124	7C	XCFPDatL	2	binary	Length of path data entry
126	7E	XCFPDatN	2	binary	Number of path data entries
128	80	XCFPDatO	4	binary	Offset to first path data entry
132	84	XCFSDatL	2	binary	Length of system data entry
134	86	XCFSDatN	2	binary	Number of system data entries
136	88	XCFSDATO	4	binary	Offset to first system data entry

XCF group entry section:
> For detail information about all fields contained in the XCF Group Entry Section, see Member Data Section in *z/OS MVS System Management Facilities (SMF)*.

XCF path entry section:
> For detail information about all fields contained in the XCF Path Entry Section, see Path Data Section in *z/OS MVS System Management Facilities (SMF)*.

XCF system entry section:
> For detail information about all fields contained in the XCF System Entry Section, see System Data Section in *z/OS MVS System Management Facilities (SMF)*.

ERBXMHG3 - Moved samples header control block

Dec offset	Hex offset	Name	Length	Format	Description
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 data set within the DSNC3 data set names table
58	3A	*	2	*	Reserved
60	3C	XMHDSACL	8	EBCDIC	Time of the last SSH available on the active data set

ERBZFXG3 - zFS performance data table

Dec offset	Hex offset	Name	Length	Format	Description
ZFXG3 header section:					
0	0	ZFXAcr	5	EBCDIC	Acronym 'ZFXG3'
5	5	ZFXVer	1	binary	ZFXG3 version X'01'
6	6	ZFXHdrL	2	binary	Length of ZFXG3 header
8	8	ZFXTotL	4	binary	Total length of ZFXG3
12	C	ZFXSysN	8	EBCDIC	System name
20	14	ZFXGDatL	4	binary	Length of general data
24	18	ZFXGDatN	4	binary	Number of general data sections
28	1C	ZFXGDatO	4	binary	Offset to general data
32	20	ZFXFDatL	4	binary	Length of file system data entry
36	24	ZFXFDatN	4	binary	Number of file system entries
40	28	ZFXFDatO	4	binary	Offset to file system data sections

44	2C	ZFXIntf	4	binary	Interface flags:
					
					Bit
					    Meaning when set
					0–7
					    Reserved
					8
					    No Locking statistics
					9
					    No User cache statistics
					10
					    No I/O count statistics
					11–12
					    Reserved
					13
					    No I/O by DASD size value
					14
					    No I/O by DASD statistics
					15
					    No Kernel statistics
					16
					    No Metadata cache statistics
					17
					    No Vnode cache statistics
					18–23
					    Reserved
					24
					    No statistics from FSINFO
					25
					    No mount size
					26
					    No mount point
					27–31
					    Reserved

Dec offset	Hex offset	Name	Length	Format	Description
48	30	ZFXStat	2	binary	Status flags: **Bit** **Meaning when set** **0** No ZFX data collected (data from previous release or option not active) **1** OMVS inactive **2** ZFS inactive or shutting down **3** Problems with one or more zFS interfaces **4** Info because aggregate is quiesced **5** Subtask timed out **6–15** Reserved
50	32	ZFXReset	2	binary	Reset flags: **Bit** **Meaning when set** **0** Locking stats reset **1** User cache stats reset **2** I/O count stats reset **3** I/O by aggregate stats reset **4** I/O by DASD stats reset **5** Kernel stats reset **6** Meta Cache stats reset **7** Vnode cache stats reset **8** FSINFO reset **9–15** Reserved
52	34	ZFXFlag	1	binary	ZFXG3 flags **Bit** **Meaning when set** **0** ZFXG3 converted to lower service level **1** ZFXG3 converted to higher release or service level **2–7** Reserved
53	35	*	3	*	Reserved
Reset time section:					
56	38	ZFX_Locking_Time	8	binary	Locking stats reset time
64	40	ZFX_UCache_Time	8	binary	User cache stats reset time

ERBZFXG3 - zFS performance data table

Dec offset	Hex offset	Name	Length	Format	Description
72	48	ZFX_IOCounts_Time	8	binary	I/O count stats reset time
80	50	ZFX_IOByAggr_Time	8	binary	I/O by aggregate stats reset time
88	58	ZFX_IOByDASD_Time	8	binary	I/O by DASD stats reset time
96	60	ZFX_Kernel_Time	8	binary	Kernel stats reset time
104	68	ZFX_MCache_Time	8	binary	Metadata cache stats reset time
112	70	ZFX_VCache_Time	8	binary	Vnode cache stats reset time
General data entry - response time section:					
120	78	ZFX_Total_Waits_for_IO	8	binary	Number of waits for I/O completion
128	80	ZFX_Total_Waits_for_Locks	8	binary	Number of waits for locks
136	88	ZFX_Total_Monitored_Sleeps	8	binary	Number of waits for events to occur
144	90	ZFX_Avg_Response_Time	8	floating point	Average response time
152	98	ZFX_Avg_ClientResponse_Time	8	floating point	Average response time (client)
160	A0	ZFX_Avg_IO_Wait_Time	8	floating point	Average wait time for I/Os
168	A8	ZFX_Avg_Lock_Wait_Time	8	floating point	Average wait time for locks
176	B0	ZFX_Avg_Monitored_Sleep_Time	8	floating point	Average wait time for events to occur
184	B8	ZFX_Total_Response_Time	8	floating point	Total response time
192	C0	ZFX_Total_ClientResponse_Time	8	floating point	Total response time (client)
200	C8	ZFX_IO_Wait_Time	8	floating point	Wait time for I/Os
208	D0	ZFX_Lock_Wait_Time	8	floating point	Wait time for locks
216	D8	ZFX_Monitored_Sleep_Time	8	floating point	Wait time for events to occur
General data entry - I/O counts by type section:					
224	E0	ZFX_IOT (3)	88		I/O counts by type
224	E0	ZFX_IO_Count	8	binary	Number of I/Os
232	E8	ZFX_IO_Waits	8	binary	Number of waits
240	F0	ZFX_IO_Cancels	8	binary	Number of I/O cancels
248	F8	ZFX_IO_Merges	8	binary	Number of I/O merges
256	100	ZFX_IO_Description	54	EBCDIC	Description
310	136	*	2	*	Reserved
General data entry - kernel data section:					
488	1E8	ZFX_Total_Requests	8	binary	Number of total zFS requests
496	1F0	ZFX_Total_XcfRequests	8	binary	Number of total zFS requests via XCF
504	1F8	ZFX_Client_Total_Requests	8	binary	Number of total zFS requests (client)

Dec offset	Hex offset	Name	Length	Format	Description
512	200	ZFX_Client_Total_XcfRequests	8	binary	Number of total zFS requests via XCF (client)
General data entry - cache activity section (user cache):					
520	208	ZFX_UC_Reads	8	binary	Number of reads
528	210	ZFX_UC_Writes	8	binary	Number of writes
536	218	ZFX_UC_Reads_Faulted	8	binary	Number of faulted reads
544	220	ZFX_UC_Writes_Faulted	8	binary	Number of faulted writes
552	228	ZFX_UC_Read_Waits	8	binary	Number of read waits
560	230	ZFX_UC_Write_Waits	8	binary	Number of write waits
568	238	ZFX_UC_ReadAsyncs	8	binary	Number of asynchronous reads
576	240	ZFX_UC_Scheduled_Writes	8	binary	Number of scheduled writes
584	248	ZFX_UC_Reclaim_Writes	8	binary	Number of page reclaim writes
592	250	ZFX_UC_FSyncs	8	binary	Number of FSyncs
600	258	ZFX_UC_TotalPages	4	binary	Number of total pages in user cache
604	25C	ZFX_UC_FreePages	4	binary	Number of free pages in user cache
608	260	ZFX_UC_AllocSegments	4	binary	Number of allocated segments
612	264	ZFX_UC_SizeL	1	binary	Length of cache size value
613	265	ZFX_UC_Size	10	EBCDIC	User cache size
623	26F	ZFX_UC_StorFixed	3	EBCDIC	User cache storage fixed
626	272	*	6	*	Reserved
General data entry - cache activity section (vnode cache):					
632	278	ZFX_VC_Requests	8	binary	Number of requests
640	280	ZFX_VC_Hits	8	binary	Number of hits
648	288	ZFX_VC_Allocates	8	binary	Number of allocated nodes
656	290	ZFX_VC_Deletes	8	binary	Number of deleted nodes
664	298	ZFX_VC_Nodes	8	binary	Number of nodes
672	2A0	ZFX_VC_SSize	8	binary	Vnode structure size
680	2A8	ZFX_VC_Extended	8	binary	Number of extended vnodes
688	2B0	ZFX_VC_ExtSSize	8	binary	Extended vnode structure size
696	2B8	ZFX_VC_USS_Held	8	binary	Number of vnodes held by USS
704	2C0	ZFX_VC_Open	8	binary	Number of open vnodes
712	2C8	ZFX_VC_SizeL	1	binary	Length of vnode cache size value
713	2C9	ZFX_VC_Size	10	EBCDIC	Vnode cache size
723	2D3	*	5	*	Reserved
General data entry - cache activity section (metadata cache):					
728	2D8	ZFX_MC_Requests	8	binary	Number of requests
736	2E0	ZFX_MC_Hits	8	binary	Number of hits
744	2E8	ZFX_MC_Updates	8	binary	Number of updates
752	2F0	ZFX_MC_PartialWrites	8	binary	Number of partial writes
760	2F8	ZFX_MC_Buffers	8	binary	Number of metadata cache buffers

ERBZFXG3 - zFS performance data table

Dec offset	Hex offset	Name	Length	Format	Description
768	300	ZFX_MC_SizeL	1	binary	Length of cache size value
769	301	ZFX_MC_Size	10	EBCDIC	Metadata cache size
779	30B	ZFX_MC_StorFixed	3	EBCDIC	Metadata storage fixed
782	30E	*	6	*	Reserved
File system data entry section:					
0	0	ZFX_FS_Name	44	EBCDIC	File system name
44	28	ZFX_FS_MountPoint	63	EBCDIC	File system mount point
107	6B	*	1	*	Reserved
108	6C	ZFX_FS_MountPoint_Len	4	binary	Mount point path length
112	70	ZFX_FS_Owner	8	EBCDIC	System name of the owner
120	78	ZFX_FS_Size	4	binary	Number of 8K blocks in aggregate
124	7C	ZFX_FS_Free	4	binary	Number of unused 8K blocks
128	80	ZFX_FS_Flags	1	binary	Flag byte: **Bit**　　**Meaning when set** **0**　　Mounted RW **1**　　Mounted RWSHARE **2**　　Aggregate not mounted **3**　　Aggregate quiesced **4–7**　　Reserved
129	81	*	7	*	Reserved
136	88	ZFX_FS_Reset	8	binary	Time statistic counters reset in seconds since last epoch
144	90	ZFX_FS_Vnodes	8	binary	Number of vnodes cached in memory
152	98	ZFX_FS_UssHeld	8	binary	Number of USS held vnodes
160	A0	ZFX_FS_Sysname	8	EBCDIC	System name these stats are for
168	A8	ZFX_FS_Open	8	binary	Number of open objects
176	B0	ZFX_FS_Tokens	8	binary	Number of tokens held from the token manager
184	B8	ZFX_FS_UserCache	4	binary	Number of 4K pages held in the user cache
188	BC	ZFX_FS_MetaCache	4	binary	Number of 8K pages held in the metadata cache
192	C0	ZFX_FS_AppReads	8	binary	Number of application reads made since last reset
200	C8	ZFX_FS_AppReadResp	8	binary	Average read response time in microseconds
208	D0	ZFX_FS_AppWrites	8	binary	Number of application writes made since last reset
216	D8	ZFX_FS_AppWriteResp	8	binary	Average write response time in microseconds
224	E0	ZFX_FS_XcfReads	8	binary	Number of XCF read calls made to the owner since last reset
232	E8	ZFX_FS_XcfReadResp	8	binary	Average XCF read call response time in microseconds
240	F0	ZFX_FS_XcfWrites	8	binary	Number of XCF write calls made to the server since last reset
248	F8	ZFX_FS_XcfWriteResp	8	binary	Average XCF write call response time in microseconds
256	100	ZFX_FS_Enospc	8	binary	Number of ENOSPC errors returned to apps since last reset

Dec offset	Hex offset	Name	Length	Format	Description
264	108	ZFX_FS_IoErrs	8	binary	Number of disk IO errors since last reset
272	110	ZFX_FS_CommErrs	8	binary	Number of XCF communication timeouts since last reset
280	118	ZFX_FS_Cancels	8	binary	Number of canceled operations since last reset
288	120	ZFX_FS_AggrReadKBytes	8	binary	Number of kBytes read from aggregate
296	128	ZFX_FS_AggrWriteKBytes	8	binary	Number of kBytes written to aggregate
304	130	ZFX_FS_Total_AppRd_RT	8	floating point	Total application read response time
312	138	ZFX_FS_Total_AppWr_RT	8	floating point	Total application write response time
320	140	ZFX_FS_Total_XcfRd_RT	8	floating point	Total XCF read response time
328	148	ZFX_FS_Total_XcfWr_RT	8	floating point	Total XCF write response time

Appendix A. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

165

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/ software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming Interface Information

This book documents intended programming interfaces that help customers to write their own z/OS Data Gatherer exit routines and to call data gatherer functions from their own applications.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Java™ is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Windows, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary contains chiefly definitions of terms used in this book, but some more general z/OS Data Gatherer, RMF, and z/OS terms are also defined.

Words that are set in *italics* in the definitions are terms that are themselves defined in the glossary.

APPC/MVS
Advanced program-to-program communication

ASCH address space
APPC transaction scheduler address space

AS
Address space

address space
That part of z/OS 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.

background session
A z/OS Data Gatherer 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.

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

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 z/OS 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 z/OS Data Gatherer, the time at the end of which one sample is taken. Varies between 50 ms and 9999 ms. See also *sample*.

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

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.

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

execution velocity
A measure of how fast work should run when ready, without being delayed for processor or storage access.

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.

GTF
generalized trace facility

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

IBM Z® Application Assist Processor (zAAP)
A special purpose processor configured for running Java programming on selected zSeries machines.

IBM Z Integrated Information Processor (zIIP)
A special purpose processor designed to help free-up general computing capacity and lower overall total cost of computing for selected data and transaction processing workloads for business intelligence (BI), ERP and CRM, and selected network encryption workloads on the mainframe.

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.

interactive session
In RMF, a monitor display-session that is controlled from the display terminal. Contrast with *background session*.

JES
Job Entry Subsystem

LCU
Logical control unit. Logical control units are also called 'Control Unit Headers ' (CUH). For details about LCU/CUH please refer to the applicable *IBM Z Input/Output Configuration Program User's Guide for ICP IOCP* (SB10-7037).

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.

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.

MPL
Multiprogramming level

OMVS
Reference to z/OS UNIX System Services

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
The activity which monitors and allocates data processing resources to applications according to goals defined in a service level agreement or other objectives.

The discipline that encompasses collection of performance data and tuning of resources.

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*.

range
The time interval you choose for your report.

Resident time
The time the address space was swapped in, in units of seconds.

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 z/OS 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 z/OS, z/VM®, and z/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 z/OS systems.

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 z/OS, a time-sharing system accessed from a terminal that allows user access to z/OS system services and interactive facilities.

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.

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

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

XCF
 Cross-system coupling facility

XES/CF
 See *Cross-system Extended Services/Coupling Facility*.

zAAP
 see IBM Z Application Assist Processor.

zIIP
 see IBM Z Integrated Information Processor.

Index

Numerics

64-bit mode
    calling sysplex data services in 22

A

about this document xi
accessibility
    contact IBM 163
answer area 41
answer area of callable services 41
assistive technologies 163

C

cache data information table, ERBCATG3 74
callable services answer area layout 41
callable services for sysplex data 21
command
    syntax diagrams xi
common answer area header layout 41
contact
    z/OS 163
coupling facility information table, ERBCFIG3 75
CPU data block 94
CPUG3_AC 97
CPUG3_HDRL 97
CPUG3_LOGITI 97
CPUG3_NUMPRC 97
CPUG3_NUMPRCOL 98
CPUG3_NUMVECOL 98
CPUG3_PHYSTI 97
CPUG3_PRCON 98
CPUG3_STATUS 98
CPUG3_TOTL 97
CPUG3_VE 97

D

data collection
    Postprocessor 1
    SMF record 1
data gatherer)
    SMF record mapping macro 2
data interface service for Monitor II 7
data reduction exit routines 32
Data Reduction Exit Routines 28
Data Set Decompression Interface Service
    parameter area 60
data set decompression interface service (ERB3RDEC) 60
decompression
    ERB3RDEC service module 60
device data set name list table 101

E

ECDFCN 109
ECDSUBN 109
ECDSUBT 109
EDECNCTN 109
EDECOLL 109
EDECORR 109
EDEG3 108
EDELU 108
EDENET 108
EDEPCKG 108
EDEPLAN 108
EDETRXC 108
EDETRXN 108
EDEUSER 108
ENCARPG 106
ENCARRAY 106
ENCCLX 106
ENCCRPG 106
ENCDECCA 107
ENCDECOM 107
ENCDECPU 107
ENCDESHP 107
ENCDESTG 107
ENCDETOT 107
ENCDEXMM 107
ENCDMN 106
ENCG3 106
ENCG3ACR 106
ENCG3DEL 106
ENCG3DEN 106
ENCG3DEO 106
ENCG3EDO 107
ENCG3HDR 106
ENCG3KFI 106
ENCG3LEN 106
ENCG3SMP 107
ENCG3TEL 106
ENCG3TEN 106
ENCG3TEO 106
ENCG3TET 106
ENCG3TLN 106
ENCG3VER 106
ENCIDLES 107
enclave data table, ERBENCG3 106
ENCNRPG 106
ENCPER 106
ENCPGN 106
ENCSRPG 106
ENCTOKEN 106
ENCUNKNS 107
ENCURPG 106
ENCUSCPU 107
ENCUSTOT 107
ERB2XD64 21
ERB2XD64 data section layout 46

IBM.®

Product Number: 5650-ZOS