z/OS
Version 2 Release 4

Resource Measurement Facility Reporter Programmer's Guide

GC27-4937-40

Note

Before using this information and the product it supports, read the information in “Notices” on page 199.

This edition applies to Version 2 Release 4 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2019-11-20

Contents

Figures

Tables

About this document

The Resource Measurement Facility (RMF) is a performance management tool that measures selected areas of system activity and presents the data collected in the form of System Management Facility (SMF) records, formatted printed reports, or formatted display reports. You can use this data to evaluate system performance and identify reasons for performance problems.

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

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

Who should use this document

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

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

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 Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

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

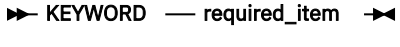
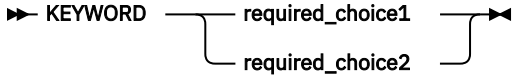
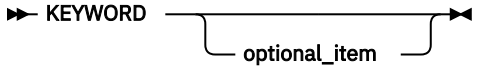
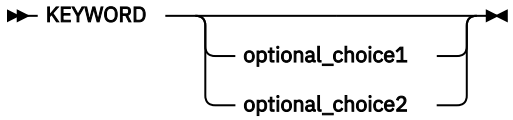
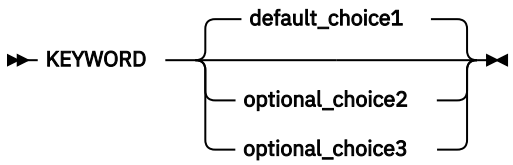
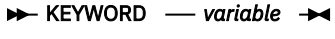
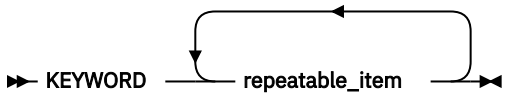
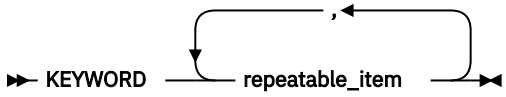
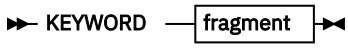
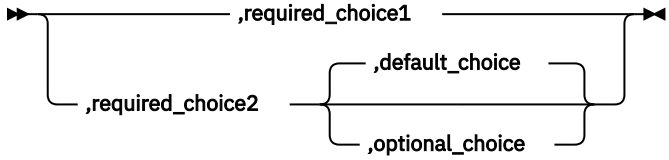
Item	Syntax example
Required item. Required items appear on the main path of the horizontal line. You must specify these items.	
Required choice. A required choice (two or more items) appears in a vertical stack on the main path of the horizontal line. You must choose one of the items in the stack.	
Optional item. Optional items appear below the main path of the horizontal line.	

Table 1. Syntax examples (continued)

Item	Syntax example
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.	
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.	
Variable. Variables appear in lowercase italics. They represent names or values.	
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.	 
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.	 **fragment** 

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 “If you have a technical problem” on page xv.

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® Knowledge Center function
If your comment or question is about the IBM Knowledge Center functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Knowledge Center Support at ibmkc@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 RMF Reporter Programmer's Guide, GC27-4937-40
- 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.

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 reporting. For programming information specific to RMF data gathering, see *z/OS RMF Data Gatherer Programmer's Guide*, GC27-4935.

New

The following information is added in this publication:

- Entries have been added to the table in "IBMz_ComputerSystem" on page 44.
- A new field was added to "PCIE – Tabular report data table ERBPCIT3" on page 169.
- Information about Extended Asynchronous Data Mover (EADM) has been added in "EADM - Tabular report data table ERBSCMT3" on page 162.
- Added "SYSRG - Tabular report data table ERBSRGT3" on page 183.

Changed

The following information is changed in this publication:

- Table fields have been updated in "CACHDET - Tabular report data table ERBCADT3" on page 145 and "CACHSUM - Tabular report data table ERBCAST3" on page 148.
- Updated SYSPLEX in Figure 2 on page 3.
- Added H, N, Z, 6, and 7 to z/OS list types. See "The listtype parameter" on page 11.
- Added SYSRG to see "RMF Phase driver table ERBPHDS3" on page 193.
- Updated tables in "CRYPTO - Tabular report data table ERBCRYT3" on page 157.
- The "PCIE – Tabular report data table ERBPCIT3" on page 169 has been updated with a new report type.
- The following tables have updated field descriptions to support restricted use common service area (RUCSA):
  - "STORC - Tabular report data table ERBCSUT3" on page 175
  - "STORCR - Tabular report data table ERBCRST3" on page 175
  -

Deleted

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

Chapter 1. Accessing performance data using the RMF Distributed Data Server

With the two flavours of the Distributed Data Server (GPMSERVE and GPM4CIM, also known as RMF XP), RMF offers a solution to monitor the performance of systems in heterogeneous environments. Application programs which want to retrieve performance data can use the HTTP API of the Distributed Data Server (DDS). The DDS returns the requested performance data as a structured XML document.

- For systems in a z/OS environment, the DDS GPMSERVE component gathers data from the RMF instances running on the sysplex members. An application program can request selected performance metrics or complete reports from the DDS. Thus, exploiters of this HTTP API have instant access to a great variety of z/OS performance metrics, including short-term information as well as long-term historical data.
- The HTTP API of the DDS can also serve requests for AIX® and Linux performance data, which are directed against an active GPM4CIM instance. RMF XP supports the following operating systems:
  - AIX on System p
  - Linux on System x
  - Linux on System z®

  GPM4CIM is the core component of RMF XP. For detailed information about RMF XP, see *z/OS RMF Reporter User's Guide*.

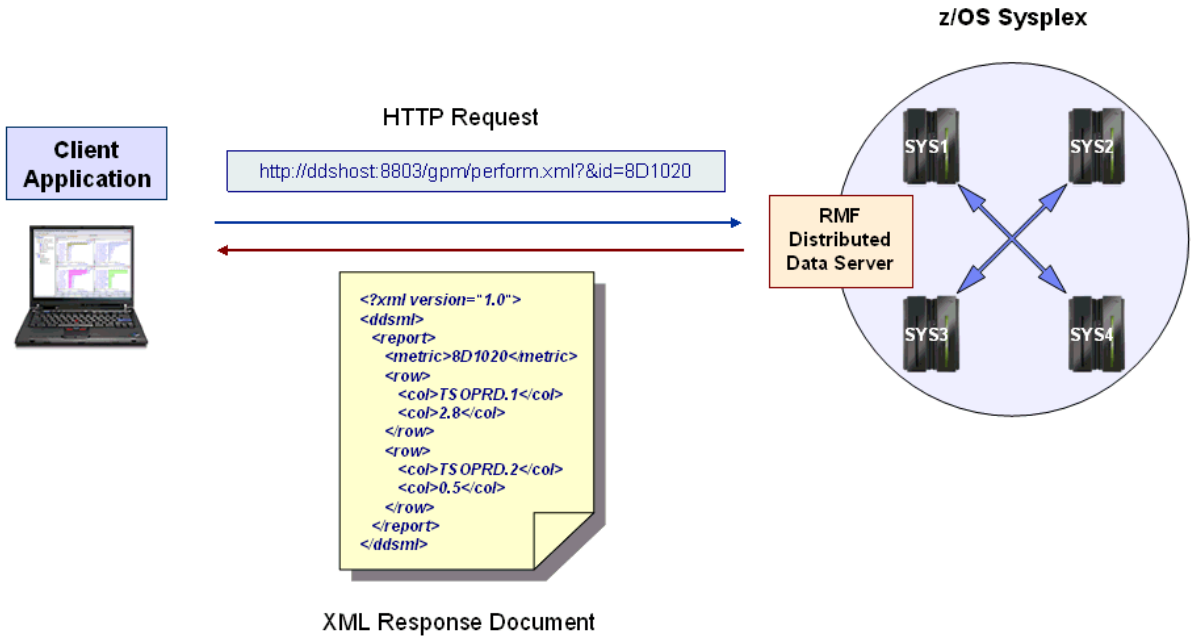


Figure 1. Example: How to use the DDS HTTP API in a z/OS environment

Figure 1 on page 1 illustrates how to use the DDS HTTP API in a z/OS environment, where requests must be sent to the GPMSERVE component of the DDS. Likewise, in a heterogeneous AIX/Linux environment, you send your HTTP request to the corresponding GPM4CIM instance of the DDS. Each instance of GPMSERVE or GPM4CIM uses a unique port to listen for incoming requests. The returned XML documents have the same syntax for both Distributed Data Server components.

This information unit describes the format and usage of DDS requests as well as the format of the returned XML documents in the following topics:

- “How to interpret an XML document returned by the DDS” on page 20

How to specify HTTP requests to the DDS for performance data

You can specify a variety of requests for different purposes. For example, you can request:

- the children of a resource
- a metric value for a resource
- a list of associated metrics for a resource
- a list of associated details for a resource
- a selection of Monitor III and Postprocessor reports

The required information how to specify such requests is contained in the following topics:

- “Understanding the underlying resource models” on page 2
- “Structure of DDS requests” on page 5
- “Description and purpose of parameters” on page 6
- “How to specify different types of requests” on page 16

Understanding the underlying resource models

It is useful to understand the concept of the used resources and their associated metrics, because most requests are specified against a resource from which you want to retrieve performance data.

The hierarchies of resources in the supported environments are illustrated in the following topics:

- “The z/OS RMF Monitor III resource model” on page 2
- “The AIX resource model” on page 4
- “The Linux on System x resource model” on page 4
- “The Linux on System z resource model” on page 5

The z/OS RMF Monitor III resource model

The z/OS RMF Monitor III resource model represents a composition of resources with the SYSPLEX as top-level resource. All other resources are children or grand-children of the SYSPLEX, for example, ***PROCESSOR*** and ***STORAGE*** resources are children of an ***MVS Image*** and grandchildren of a ***SYSPLEX***. The entire hierarchical model looks as shown in Figure 2 on page 3.

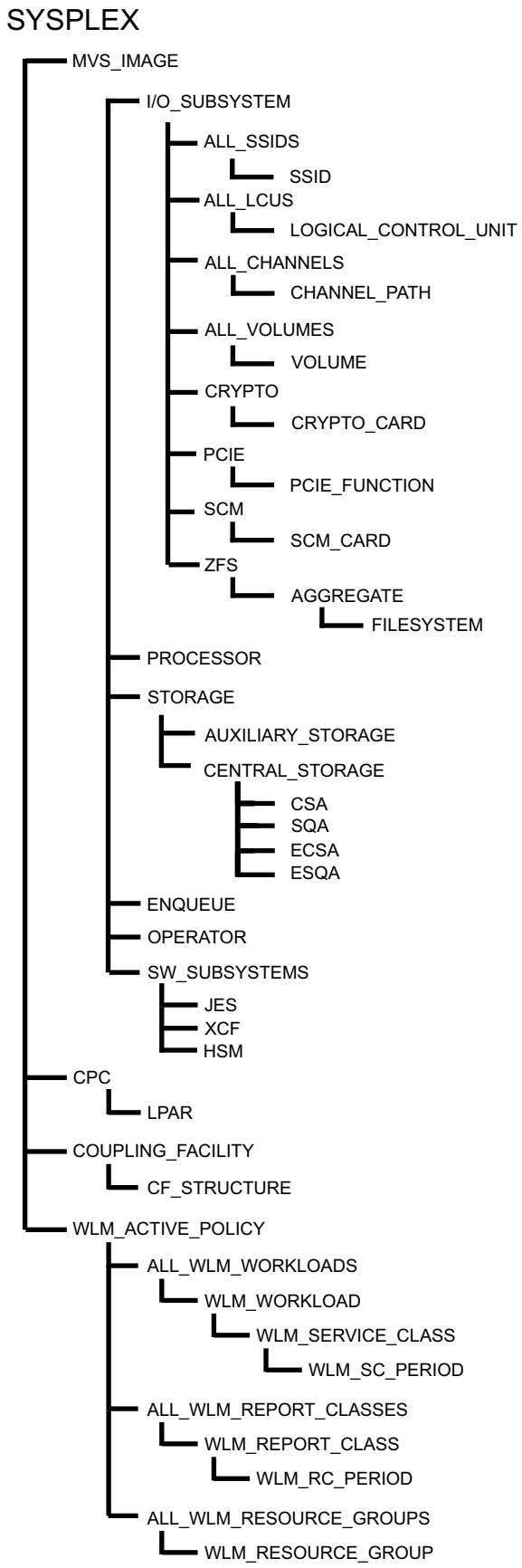


Figure 2. The z/OS RMF Monitor III resource model

The AIX resource model

The AIX resource model represents a composition of resources with an AIX_SYSTEM_COMPLEX as top-level resource. An AIX_SYSTEM_COMPLEX resource denotes a complex of distributed systems running AIX, where the performance data is gathered by a CIM server. All other resources are children or grand-children of the AIX_SYSTEM_COMPLEX. The entire hierarchical model looks as shown in Figure 3 on page 4.

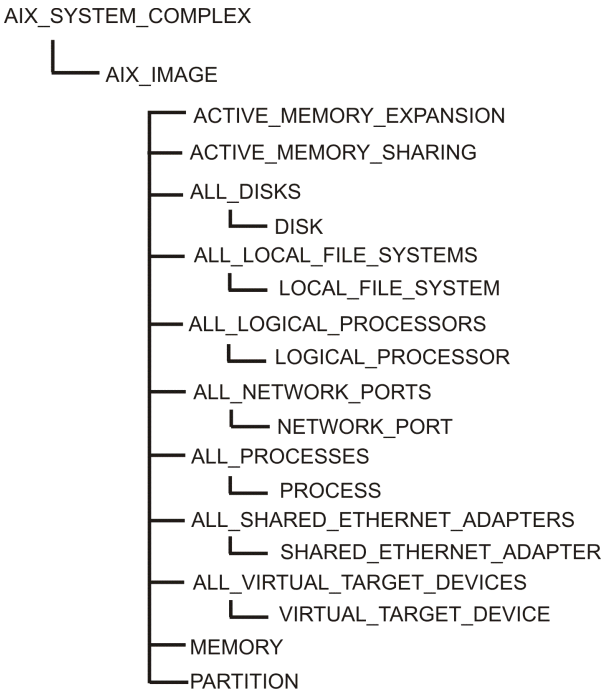


Figure 3. The AIX resource model

The Linux on System x resource model

The Linux on System x resource model represents a composition of resources with an XLINUX_SYSTEM_COMPLEX as top-level resource. An XLINUX_SYSTEM_COMPLEX resource denotes a complex of distributed systems running Linux on System x, where the performance data is gathered by a CIM server. All other resources are children or grand-children of the XLINUX_SYSTEM_COMPLEX. The entire hierarchical model looks as shown in Figure 4 on page 5.

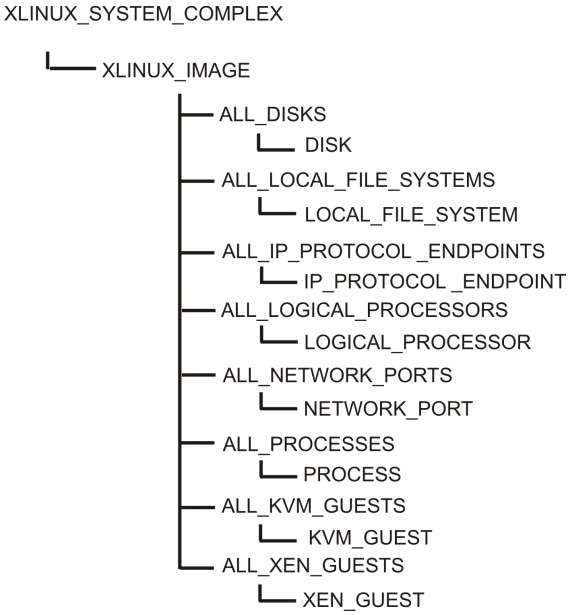


Figure 4. The Linux on System x resource model

The Linux on System z resource model

The Linux on System z resource model represents a composition of resources with a ZLINUX_SYSTEM_COMPLEX as top-level resource. A ZLINUX_SYSTEM_COMPLEX resource denotes a complex of distributed systems running Linux on System z, where the performance data is gathered by a CIM server. All other resources are children or grandchildren of the ZLINUX_SYSTEM_COMPLEX. The entire hierarchical model looks as shown in Figure 5 on page 5.

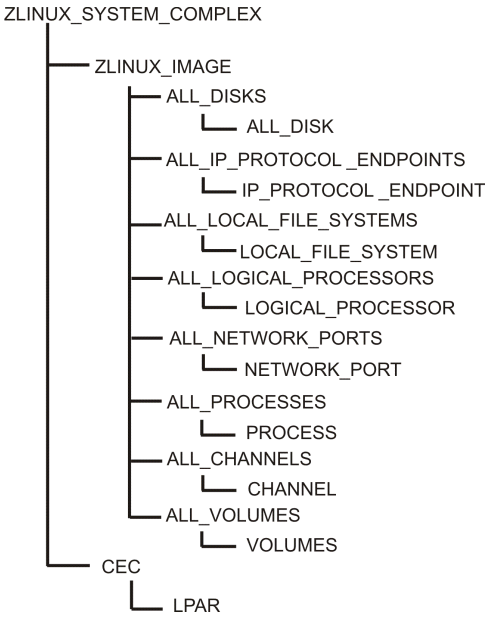


Figure 5. The Linux on System z resource model

Structure of DDS requests

To request performance data in a z/OS environment, an application must send an HTTP request to the DDS server located on the monitored z/OS sysplex.

To request cross platform performance data from distributed platforms, the HTTP request must be sent to the DDS server with the monitored AIX or Linux system defined in its configuration file.

Topic “Description and purpose of parameters” on page 6 first explains the set of request parameters, used in the various request types. Then, “How to specify different types of requests” on page 16 presents detailed information about the purpose of the request types and how to specify them.

Here is an example request for a certain performance metric for a specified resource: it requests the response time (denoted by the metric ID ***8D10F0***) of volume ***SYSLIB*** of system ***SYSA***:

```
http://ddshost:8803/gpm/perform.xml?resource=SYSA,SYSLIB,VOLUME&id=8D10F0
```

An example request for the Postprocessor CPU and CRYPTO reports looks similar to the following:

```
http://ddshost:8803/gpm/rmfpp.xml?reports=CPU,CRYPTO&date=20090801,20090804
```

General request syntax:

```
http://<ddshost>:<ddsport>/gpm/<filename>?<parm_1>=<value_1>&…&<parm_n>=<value_n>
```

The parameters have the following meanings:

ddshost
is the IP address or the symbolic name of the DDS server.

Example:

```
ddshost
```

ddsport
is the port number of the DDS server (GPMSERVE or GPM4CIM).

Example:

```
8803
```

filename
is the filename of the XML document you want to retrieve, followed by the extension `.xml`. It determines the request type and the returned XML document type. The valid filenames are described in “How to specify different types of requests” on page 16.

Examples:

```
perform.xml
rmfpp.xml
```

<parm_1>=<value_1>&...&<parm_n>=<value_n>
is the query string within the request. It is composed of multiple parameter/value pairs, separated by **&** characters, that precisely specify the requested information. Available parameters are described in “Description and purpose of parameters” on page 6.

Description and purpose of parameters

Table 2 on page 6 shows an overview of parameters which you can specify in the query string of the request. The remainder of this topic describes the parameters in detail.

Table 2. Request parameters

Parameter	Purpose
Parameters for Monitor III and CIM requests	
resource	Monitor III resource identifier
id	metric identifier associated with the resource
range	start and end time of data interval

Table 2. Request parameters (continued)	
Parameter	**Purpose**
filter	filter string
workscope	workscope specification
name	name of list element
listtype	type of list element
report	Monitor III report name
Parameters for Postprocessor requests	
reports	list of Postprocessor report names
date	start and end date of the reporting period for the requested Postprocessor report(s)
duration	interval length for the requested Postprocessor duration report(s)
timeofday	start and end time of the reporting period for each day in the reporting period
sysid	system name for single system reports
overview	list of control statements for the Overview report
timeout	timeout period in seconds for the completion of Postprocessor jobs
joboutdel	indication if job output is to be deleted after being processed successfully
smfdata	list of SMF input data
sortsmf	indication if sorting of SMF data is necessary

The resource parameter

`resource`=[ulq],[resource_name],resource_type

This parameter describes the resource for which information is requested. The **`resource`** parameter is composed of three parts:

- upper level qualifier (ULQ)
- resource name
- resource type

You can see the available resource types in the syntax required for the request string in Figure 2 on page 3.

An ULQ is needed for the resource parameter, because resources with the same name can exist multiple times in a sysplex, for example volumes or channels. For most of the resources, the ULQ is the name of the associated z/OS system.

For the sysplex resource, the ULQ can be omitted. In such a case, the resource specification starts with a comma. For unique resources like the PROCESSOR resource in an MVS IMAGE, you can either omit the resource name or you can assign an asterisk as the resource name.

Examples for Monitor III resources:

- An MVS image named ***SYSA*** in a sysplex is represented as `resource=,SYSA,MVS_IMAGE`. That is, in this case the upper level qualifier can be omitted, since the system name is unique in a sysplex.
- An instance of resource type CHANNEL_PATH with ID ***0F*** in MVS_IMAGE ***SYSA*** is represented as `resource=SYSA,0F,CHANNEL_PATH`. Here, all three parts of the resource identifier are required, because channel paths are not unique in a sysplex.

- An instance of resource type PROCESSOR, which is unique in MVS_IMAGE ***SYSA*** is represented as `resource=SYSA,,PROCESSOR` or `resource=SYSA,*,PROCESSOR`.

Examples for Linux/AIX resources: (resource names are case-sensitive)

- A Linux on System z image named ***LINZRMF5*** is represented as `resource=,LINZRMF5,ZLINUX_IMAGE`. You can omit the upper level qualifier, as system names are unique in a ZLINUX_SYSTEM_COMPLEX, and also are unique in an XLINUX_SYSTEM_COMPLEX and in an AIX_SYSTEM_COMPLEX, too.
- An AIX image named ***p6rmf2*** is represented as `resource=,p6rmf2,AIX_IMAGE`.
- A process on a Linux image named ***LINXRMF5*** with resource name ***102*** is represented as `resource=LINXRMF5,102,PROCESS`. Here, all three parts of the resource identifier are required, because processes are not unique in Linux/AIX complexes.

  `resource=p6rmf2,cimlistener[7209212],PROCESS` is an example of a process on an AIX image.
- An instance of resource type LOGICAL_PROCESSOR with resource name ***CPU0*** on system ***P6RMF1*** is represented as `resource=P6RMF1,CPU0,LOGICAL_PROCESSOR`. This notation is valid for either Linux or AIX.

The id parameter

id=metric_ID

This parameter identifies the Monitor III metric that you want to request. The metric identifier must be applicable for the resource given in the same request. You can retrieve a metric ID in the following ways, depending on whether you search for any arbitrary metric or for a metric that is associated with a certain resource:

1. To find any arbitrary metric ID, search the file `http://<ddshost>:<ddsport>/gpm/index.xml` for the desired metric. For example, searching for the metric ***# active users*** would return the desired information:

   ```
   <metric id="8D0620">
     <description># active users</description>
   …
   …
   ```

2. To find a metric ID that is associated with a certain resource (in our example for a resource of type MVS_IMAGE), specify a listmetrics request similar to the following one:

   ```
   http://<ddshost>:<ddsport>/gpm/listmetrics.xml?resource=,<system_name>,MVS_IMAGE
   ```

   The returned metric list contains the desired information in the same format as shown for alternative “1” on page 8.

The range parameter

range=YYYYMMDDhhmmss[,YYYYMMDDhhmmss]

This parameter specifies the time interval for the requested performance data. Start and end times are specified as a string of the form YYYYMMDDhhmmss, provided in local time of the monitored system. If you omit the end time, the defined Monitor III gatherer interval (MINTIME) is used to determine the end time.

Example:

```
RANGE=20090725100000,20090725110000
```

specifies the start time as 10:00 AM on 07/25/2009 and the end time as 11:00 AM on the same day.

The filter parameter

filter=list_of_filter_criteria

You can focus on the data of your interest by adding a filter specification when requesting a list of values. You can use filters to specify the following:

- one or more name patterns to be matched against the names in the list
- a lower and upper bound to be compared to the values in a list
- a maximum list length with an indicator to select the instances with either the highest or the lowest values
- a sorting order for either the names or the values of the list (ascending or descending).

You can separate multiple filter criteria by a semicolon (“;”).

The following keywords are available for filters:

PAT=<expression>
Specifies one or more patterns which must match the name part of a list element.

LB=<double>
Specifies a lower bound value. That is, only list elements with values higher than the given lower bound are returned.

UB=<double>
Specifies an upper bound value. That is, only list elements with values lower than the specified upper bound are returned.

HI=<integer>
Only the highest <integer> list elements are returned (mutually exclusive with LO). The default is 20.

LO=<integer>
Only the lowest <integer> list elements are returned (mutually exclusive with HI).

ORD=<xx>
Sort the list of name/value pairs by their names in ascending (NA) or descending (ND) order (<xx>=NA or ND), or by their values in ascending (VA) or descending (VD) order. The default is VD. If you do not want to have any sort order, specify ORD=NN.

If you do not explicitly specify a filter, the default settings are `&filter=HI=20;ORD=VD` which returns the top 20 values, sorted by value, in descending order (VD).

Examples:

Filter term	Result
&filter=PAT=*CICS*|*SH*	only instances containing the name patterns CICS®, SH, or both are returned
&filter=LB=10.5	only instances with a value higher than 10.5 are returned
&filter=UB=12.8	only instances with a value lower than 12.8 are returned
&filter=HI=5	only the instances containing the five highest values are returned
&filter=LO=5	only the instances containing the five lowest values are returned
&filter=ORD=NA	specifies that the returned instances should be sorted by name in ascending order

The workscope parameter

`workscope`=[ulq],name,type

Use this parameter to qualify a request for performance data in more detail with regard to address spaces and WLM entities. Workscopes can be applied to single valued metrics as well as to list valued metrics. For example,

- for the metric *performance index*, the **workscope** parameter denotes the associated service class period
- for the metric *% workflow by job*, you can use this parameter to focus on jobs that belong to a certain service class.

The **workscope** parameter consists of three parts:

- an upper level qualifier which may be blank or which specifies the name of a WLM service class, if the workscope type is a WLM service class period
- a workscope name (for example, job name or report class name) or a service class period
- a workscope type

Available workscope types:

G
global (no workscope required)

W
WLM workload

S
WLM service class

P
WLM service class period

R
WLM report class

J
job

Examples:

Workscope term	result
&workscope=,BATCH,W	focus on workload BATCH
&workscope=HOTTSO,3,P	focus on period 3 of service class HOTTSO
&workscope=,CICSPRD,R	focus on report class CICSPRD
&workscope=,CATALOG,J	focus on job catalog

Example request strings with a workscope parameter

Example 1 (single valued metric)
Retrieve the performance index (ID = 8D1000) for the first period of service class BATCHMED:

```
http://ddshost:8803/gpm/perform.xml?resource=,MVSPLEX,SYSPLEX
                                  &id=8D1000&workscope=BATCHMED,1,P
```

Example 2 (list valued metric)
Retrieve the workflow by job (ID = 8D0560) for all jobs running in service class HOTTSO:

```
http://ddshost:8803/gpm/perform.xml?resource=,SYSA,MVS_IMAGE
                                  &id=8D0560&workscope=,HOTTSO,S
```

The name parameter

name=resource_name

This parameter is required in a *detailsname* request, which returns the attributes of a resource. In such a request, the **name** parameter specifies the name of a resource for which the attributes are requested. For example, you can retrieve the attributes of a volume with a specified name as shown in "How to specify a detailsname request" on page 17.

The listtype parameter

listtype=type

This parameter is required for the following request types:

- in a *detailsname* type request together with the **name** parameter. It specifies the list type of the resource for which details are requested. For example, if you want to know attributes of a volume, you need to specify **listtype**=V.
- in a *filterinst* type request. In such a request, the list type denotes the resource type of the requested filter instances.

  For example, you use the **listtype** parameter to retrieve either volumes (**listtype**=V) or channels (**listtype**=C) as filter instances for the sysplex resource.

Available z/OS list types:

A
partition

B
job (only for SYSPLEX resource)

C
channel path

D
data set

E
enclave

F
coupling facility

G
zFS aggregate

H
PCIE function

I
SSID

J
job

K
WLM report class period

L
LCU

M
MVS image

N
Data set and job

O
USS pid and job

P
WLM service class period

Q
storage group

R
WLM report class

S
WLM service class

T
CF structure

U
channel path and CU

V
volume

W
WLM workload

X
CPC

Y
zFS file system

Z
SCM card

6
crypto card

7
WLM resource group

Available AIX list types:

A
shared ethernet adapter

D
disk

F
file system

I
AIX image

L
logical processor

N
network port

P
process

V
virtual target device

Available Linux on System x list types:

D
disk

E
IP protocol endpoint

F
local file system

I
Linux image

K
KVM domain

L
logical processor

N
network port

P
process

X
XEN domain

Available Linux on System z list types:

C
CEC

D
disk

E
IP protocol endpoint

F
local file system

H
channel

I
Linux image

L
logical processor

N
network port

P
process

R
LPAR

V
volume

The report parameter

report=report_name

This parameter specifies the name of one Monitor III report to be returned by the request.

Example:

```
report=PROCU
```

Requests a Monitor III Processor Usage report.

The reports parameter

reports=report_name[(options)][, report_name[(options)], ... report_name[(options)]]

This parameter enumerates one or more identifiers of Postprocessor reports to be returned by the request. You can define options for applicable reports, as described in "Long-term reporting with the Postprocessor" in *z/OS RMF Reporter User's Guide*.

Examples:

- Get detailed data for service class TSOPROD by requesting the Service Class report:

```
reports=WLMGL(SCLASS(TSOPROD))
```

- Create a Workload Activity report and assume that all CICS applications run in the three workload groups CICSPROD, CICSTEST, and CICSADMN. Get the Workload Group report for all groups:

```
reports=WLMGL(WGROUP(CICS*))
```

- Request a list of Postprocessor reports, with the WLMGL report with suboptions:

```
reports=CPU,CRYPTO,WLMGL(SCPER(STCLOW))
```

Notes:

1. The enumerated report identifiers can be enclosed in double quotes.
2. You cannot use the ***reports*** parameter and the ***overview*** parameter in the same request. That is, you can either request one or more standard Postprocessor reports or one Overview report (see "The overview parameter" on page 15).

For a complete list of available Postprocessor reports in XML format, see *z/OS RMF Reporter User's Guide*.

The date parameter

date=<start-date>,<end-date>

This parameter specifies the start and end date of the reporting period for all Postprocessor reports in the format yyyymmdd or yyddd.

Example:

```
date=20101125,20101126
date=10256,10257
```

The duration parameter

duration=<interval-length>

This parameter specifies that the Postprocessor is to generate duration reports and indicates the length of the duration interval in the format ***hhmm***. The minimum value is 0000 which is corrected by the Postprocessor to the interval length that is found in the data being processed. The maximum value is 9960 which is equivalent to 100 hours.

Example:

```
duration=1200
```

The timeofday parameter

timeofday=<start-time>,<end-time>

This parameter specifies the start and end time of the reporting period for each day in the reporting period in the format hhmm.

Example:

```
timeofday=0800,1600
```

The sysid parameter

sysid=<cccc>

This parameter identifies the single system for which the reports are to be generated. It is ignored for sysplex reports.

Example:

```
sysid=SYSA
```

The overview parameter

overview=(<statement_1>),(<statement_2>)...(<statement_n>)

where <statement_n> is

```
control_statement_name(condition_name(qualifier))[,SYSTEMS | ,NOSYSTEMS]
```

This parameter contains a list of control statements for the Overview report, equivalent to the OVW control statements as described in *z/OS RMF Reporter User's Guide*. The maximum number of control statements is 253.

Example: The request parameter

```
overview=(DATA01(CADSTG(SSID(0600),DEVN(06F3)))),
(DB2PRD(CADRT(DEVN(0722),SSID(0700)))),
(RHT0050(CASRHT(SSID(0050))))
```

represents the following overview control statements:

```
OVW(DATA01(CADSTG(SSID(0600),DEVN(06F3))))
OVW(DB2PRD(CADRT(SSID(0700),DEVN(0722))))
OVW(RHT0050(CASRHT(SSID(0050))))
```

The timeout parameter

timeout=<wait-time>

This parameter specifies the timeout period in seconds, that the DDS should wait for Postprocessor jobs to complete. The valid range is from 0 to 3600 seconds. The default value is 300 seconds.

Example:

```
timeout=1200
```

The joboutdel parameter

joboutdel=[YES | NO]

This parameter specifies whether the held output of the Postprocessor job should be deleted by the RMF Distributed Data Server after it has been processed successfully. The default is NO.

Examples:

joboutdel=NO
joboutdel=YES

The smfdata parameter

smfdata=smf_data[,smf_data_2,...,smf_data_k]

This parameter contains a list of names of SMF data sets or logstreams which are used as input for the generation of Postprocessor reports. The names must be fully qualified and valid z/OS data set names.

Examples:

smfdata=RMF.SMFDATA.SYSA,RMF.SMFDATA.SYSB,RMF.SMFDATA.SYSC
smfdata=IFASMF.PERF.SYSDPLEX

The sortsmf parameter

sortsmf=[YES | NO]

This parameter specifies whether the SMF data, which is defined with the **smfdata** parameter, is sorted before it is used as input for the generation of Postprocessor reports. The default is NO.

Examples:

sortsmf=NO
sortsmf=YES

How to specify different types of requests

This topic explains the purpose of the available types of requests and describes how to specify each request type. A request type is determined by the specified value of the ***<filename>*** parameter.

There is one subtopic for each available request type, presenting an example request and listing the required and optional parameters. The resulting XML documents are documented in "How to interpret an XML document returned by the DDS" on page 20.

Table 3 on page 16 lists all filename specifications which are valid in a request string to a z/OS system and also indicates which filenames can be specified for AIX or Linux.

Table 3. Valid filename specifications

Filename	See	Purpose	XML document type	Valid for AIX or Linux
contained.xml	"How to specify a contained request" on page 17	list of child resources	<contained-resources-list>	yes
details.xml	"How to specify a details request" on page 17	attributes of a resource	<attribute-list>	no
detailsname.xml	"How to specify a detailsname request" on page 17	attributes of a resource	<attribute-list>	no
filterinst.xml	"How to specify a filterinst request" on page 17	list of filter instances	<filter-instances-list>	yes
index.xml	"How to specify an index request" on page 18	list of metrics for all resources	<metric-list>	yes
listmetrics.xml	"How to specify a listmetrics request" on page 18	list of metrics for one resource	<metric-list>	yes
perform.xml	"How to specify a perform request" on page 18	performance data	<report>	yes
rmfm3.xml	"How to specify a report request" on page 18	Monitor III report	<report>	no

Table 3. Valid filename specifications (continued)				
Filename	**See**	**Purpose**	**XML document type**	**Valid for AIX or Linux**
rmfpp.xml	"How to specify a postprocessor request" on page 19	one or more Postprocessor reports	<postprocessor>	no
root.xml	"How to specify a root request" on page 19	root resource	<contained-resources-list>	yes
workscopes.xml	"How to specify a workscopes request" on page 19	list of workscopes	<workscope-list>	no

How to specify a *contained* request

A request using this filename returns the list of children for the specified resource. The result is an XML file of type <contained-resources-list>.

Example request strings:

```
http://ddshost:8803/gpm/contained.xml?resource=,MVSPLEX,SYSPLEX
http://ddshost:8805/gpm/contained.xml?resource=,,AIX_SYSTEM_COMPLEX
http://ddshost:8805/gpm/contained.xml?resource=,P6RMF1,AIX_IMAGE
```

Required parameter:
resource

How to specify a *details* request

A request using this filename returns the list of attributes for the selected resource. The result is an XML file of type <attribute-list>.

Example request string:

```
http://ddshost:8803/gpm/details.xml?resource=,SYSA,MVS_IMAGE
```

Required parameter:
resource

How to specify a *detailsname* request

A request using this filename returns the list of attributes for the single resource designated by the ***name*** parameter. You must also specify the type of the list, for example &listtype=V for volumes. The result is an XML file of type <attribute-list>.

Example request string:

```
http://ddshost:8803/gpm/detailsname.xml?resource=SYSA,*,ALL_VOLUMES&name=SYSLIB
                                                                  &listtype=V
```

Required parameters:
resource, *name*, *listtype*

How to specify a *filterinst* request

A request using this filename returns a list of all possible filter instances with the specified list type for the given resource. The result is an XML file of type <filter-instances-list>.

Example request strings:

```
http://ddshost:8803/gpm/filterinst.xml?resource=,MVSPLEX,SYSPLEX&listtype=C
http://ddshost:8807/gpm/filterinst.xml?resource=,ZLINUXPLEX,ZLINUX_SYSTEM_COMPLEX
                                                          &listtype=V
```

Required parameters:
resource, *listtype*

How to specify an *index* request

A request using this filename returns the list with all resources and associated metrics in the sysplex. It is invoked without any parameters. The result is an XML file of type <metric-list>.

Example request string:

```
http://ddshost:8803/gpm/index.xml
```

How to specify a *listmetrics* request

A request using this filename returns the list of associated metrics for the specified resource type. The result is an XML file of type <metric-list>.

Example request strings:

```
http://ddshost:8803/gpm/listmetrics.xml?resource=,SYSA,MVS_IMAGE
http://ddshost:8805/gpm/listmetrics.xml?resource=P6RMF1,CPU0,LOGICAL_PROCESSOR
```

Required parameter:
resource

How to specify a *perform* request

A request using this filename returns performance data for the selected metric of the specified resource. The result is an XML file of type <report>.

Example request strings:

```
http://ddshost:8803/gpm/perform.xml?resource=,SYSA,MVS_IMAGE&id=8D0160
http://ddshost:8806/gpm/perform.xml?resource=,XLINSYSA,XLINUX_IMAGE&id=203160
http://ddshost:8807/gpm/perform.xml?resource=ZLINXRMF,PROC0,LOGICAL_PROCESSOR
                                                          &id=304010
```

Required parameters:
resource, *id*

Optional parameters:
range, *filter*, *workscope*

How to specify a *report* request

A request using this filename returns a complete RMF Monitor III report for the specified resource. The result is an XML file of type <report>.

The request string for a Monitor III report contains the report name, such as CHANNEL or SYSSUM, specified with the report parameter.

Example request string:

```
http://ddshost:8803/gpm/rmfm3.xml?report=CHANNEL&resource=,SYSA,MVS_IMAGE
```

Required parameters:
report, *resource*

Optional parameters
range

- To request a list of available reports with sysplex scope, specify a listmetrics request for the SYSPLEX resource, for example:

```
http://ddshost:8803/gpm/listmetrics.xml?resource=,MVSPLEX,SYSPLEX
```

- To request a list of available reports with single system scope, specify a listmetrics request for an MVS_IMAGE resource, for example:

```
http://ddshost:8803/gpm/listmetrics.xml?resource=,SYSA,MVS_IMAGE
```

Available reports are listed in the returned document as follows:

```
<metric id="report_name">
<format>report</format>
</metric>
```

How to specify a *root* request

A request using this filename returns the z/OS SYSPLEX, or the respective Linux or AIX SYSTEM_COMPLEX as root resource. It is invoked without any parameters. The result is an XML file of type <contained-resources-list> containing only the root resource.

Example request string:

```
http://ddshost:8803/gpm/root.xml
```

How to specify a *workscopes* request

A request using this filename returns the list of associated workscopes for the specified resource. The result is an XML file of type <workscope-list>.

- A request against a SYSPLEX resource returns all WLM entities in the sysplex.
- A request against an MVS_IMAGE resource returns the active jobs in the system.

Example request string:

```
http://ddshost:8803/gpm/workscopes.xml?resource=,MVSPLEX,SYSPLEX
```

Required parameter:
resource

How to specify a postprocessor request

A request using filename ***rmfpp.xml*** returns either the requested (list of) standard Postprocessor report(s) or an Overview report, depending on the parameters (see “Description and purpose of parameters” on page 6). The result of the request is an XML file of type <postprocessor> containing the requested report(s).

Example request string:

Request for a list of reports containing the ***CPU*** and the ***CRYPTO*** report between the first and fourth of August 2010:

```
http://ddshost:8803/gpm/rmfpp.xml?reports=CPU,CRYPTO&date=20100801,20100804
```

Required parameter:
either **`reports`** or **`overview`**

Optional parameters:
`date`, **`duration`**, **`timeofday`**, **`sysid`**, **`timeout`**, **`joboutdel`**, **`smfdata`**, **`sortsmf`**

How to interpret an XML document returned by the DDS

This topic contains all syntax information needed to read the XML documents returned by the DDS. It describes the syntax rules of all XML tags used in the returned documents. From these documents, your application program can extract the requested performance data.

The data types **token**, **NMTOKEN**, **byte**, and **nonNegativeInteger** of the returned values are used as defined in the XML Schema language.

Description of the XML document structure

Each XML document type starts with the <ddsml> tag, followed by the <server>...</server> specification.

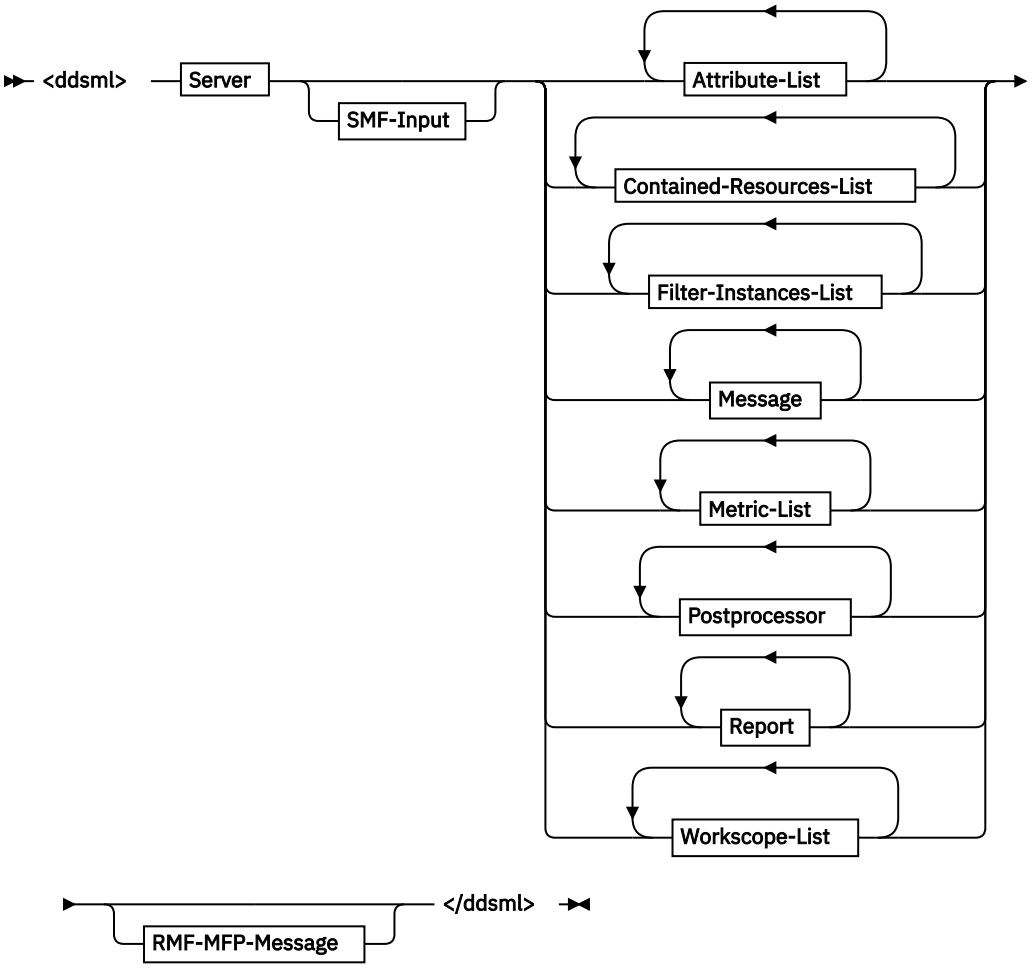
The content which follows the <server>...</server> tags, is enclosed into a pair of corresponding start and end tags, for example:

```
<attribute-list> … content … </attribute-list>
```

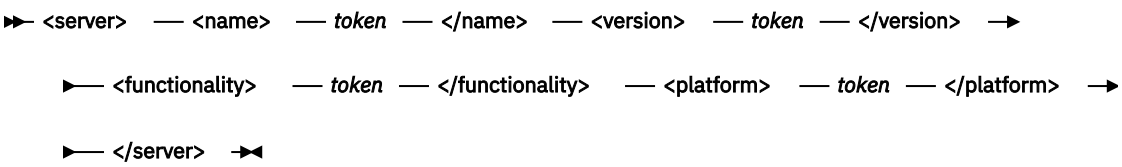
or:

```
<report> … content … </report>
```

Syntax: ddsml



Server



Tag/Fragment	Description	Type
server	characteristics of the DDS server	see syntax diagram
name	name of the DDS server, value is set to ***RMF-DDS-Server***	token
version	operating system release	token
functionality	level number of the DDS server	token
platform	operating system	token
SMF-Input	list of SMF input data	See "SMF-Input element" on page 22
Attribute-List	attributes for a resource	See "Attribute-List element" on page 22
Contained-Resources-List	list of child resources	See "Contained-Resources-List element" on page 24
Filter-Instances-List	list of filter instances	See "Filter-Instances-List element" on page 25
Message	error message or warning	See "Message" on page 26
Metric-List	list of metrics for a resource	See "Metric-List element" on page 26
Postprocessor	one or more Postprocessor reports	See "Postprocessor element" on page 28
Report	performance data	See "Report element" on page 34
Workscope-List	list of workscopes	See "Workscope-List element" on page 38
RMF-MPF-Message	RMF Postprocessor MFPMSG error messages or warnings	See "RMF-MFP-Message element" on page 39

As mentioned in "How to specify different types of requests" on page 16, the document type is determined by the ***filename*** specification in the request. A specification of the same document type may occur multiple times in one XML document, as shown in the following example, where there are multiple instances of document type ***metric***.

Example:

```
<ddsml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="/gpm/include/ddsml.xsd">
<server>
<name>RMF-DDS-Server</name>
<version>ZOSV2R4</version>
<functionality>2381</functionality>
<platform>z/OS</platform>
```

```
</server>
<metric-list>
<resource>
…
</resource>
<metric id="8D2060">
…
</metric>
…
…
…
<metric id="8D2170">
…
</metric>
</metric-list>
</ddsml>
```

The remainder of this topic describes the syntax of the available document types and presents an example document for each type.

SMF-Input element

A <smf-input> tag provides information on the names of the SMF input data sets or logstreams used for the generated Postprocessor report.

Syntax: SMF-Input

Tag/Fragment	Description	Type
file	name of a SMF data set or logstream	token

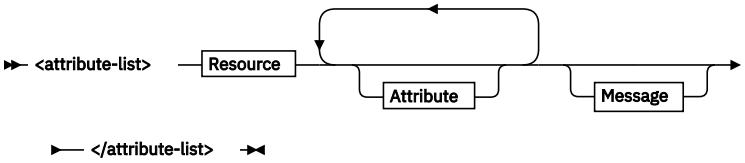
Example XML code:

```
<smf-input>
   <file>RMF.SMFDATA.SYSA</file>
   <file>RMF.SMFDATA.SYSB</file>
</smf-input>
```

Attribute-List element

The <attribute-list> tag lists a resource and its attributes.

Syntax: Attribute-List



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23
Attribute	list of attributes	See “Attribute” on page 24
Message	error message or warning	See “Message” on page 26

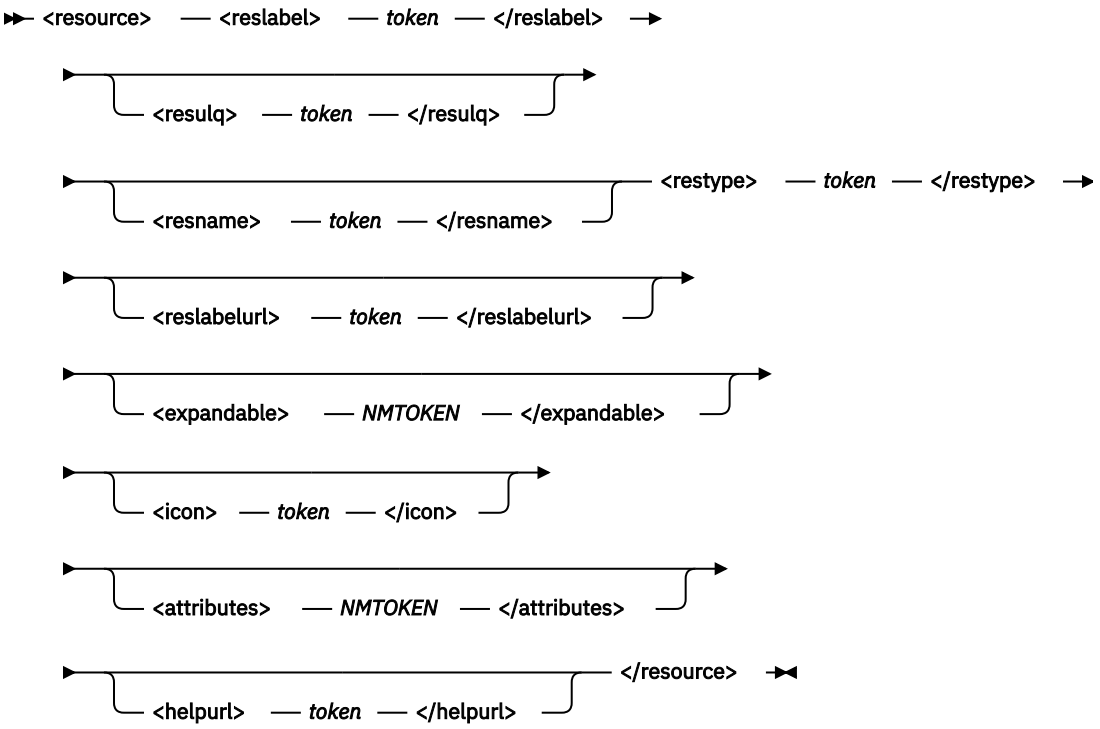
Example XML code:

```
<attribute-list>
<resource>
…
</resource>
<attribute id="37">
…
</attribute>
…
<attribute id="78">
…
</attribute>
</attribute-list>
```

Resource

The <resource> tag specifies a resource in detail.

Syntax: Resource



Tag/Fragment	Description	Type
reslabel	description of resource	token
resulq	upper level qualifier of the resource	token
resname	resource name	token
restype	resource type	token
reslabelurl	resource label for use in an URL without blanks	token
expandable	denotes whether resource itself contains other resources; value in {YES, NO}	NMTOKEN
icon	name of icon image for this resource	token
attributes	denotes whether attributes may be queried for this resource; value in {YES, NO}	NMTOKEN

Tag/Fragment	Description	Type
helpurl	URL of help description	token

Example XML code:

```
<resource>
<reslabel>,CF01,COUPLING_FACILITY</reslabel>
<resname>CF01</resname>
<restype>COUPLING_FACILITY</restype>
<reslabelurl>,CF01,COUPLING_FACILITY</reslabelurl>
<expandable>YES</expandable>
<icon>rmfcf.gif</icon>
<attributes>YES</attributes>
</resource>
```

Attribute

The <attribute> tag specifies the name and value of each attribute of a resource.

Syntax: Attribute

Tag/Fragment	Description	Type
id	unique attribute ID	token
description	description of a single attribute	token
value	value of this attribute	token

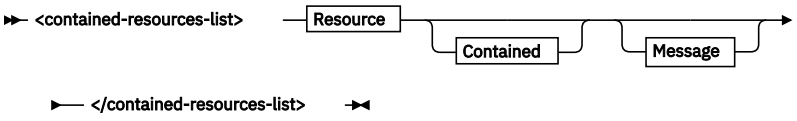
Example XML code:

```
<attribute id="1">
<description>Processor type</description>
<value>2064</value>
</attribute>
<attribute id="16">
<description>unit</description>
<value>3390-9</value>
</attribute>
```

Contained-Resources-List element

The <contained-resources-list> tag lists all child resources of a given resource within the resource tree.

Syntax: Contained-Resources-List



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23
Contained	list of child resources	See “Contained” on page 25

Tag/Fragment	Description	Type
Message	error message or warning	See “Message” on page 26

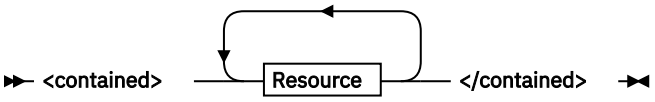
Example XML code:

```
<contained-resources-list>
<resource>
…
</resource>
<contained>
…
</contained>
</contained-resources-list>
```

Contained

The <contained> tag encloses the list of child resources.

Syntax: Contained



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23

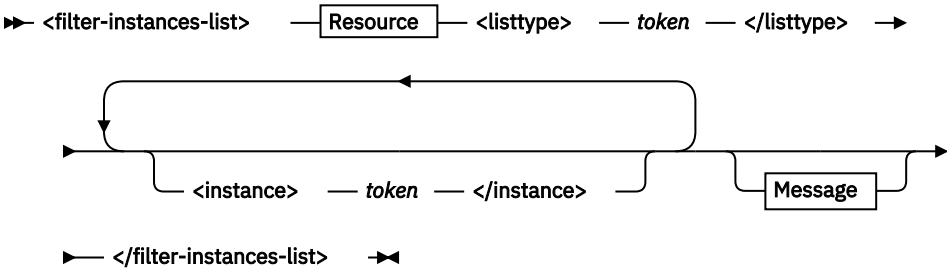
Example XML code:

```
<contained>
<resource>
<reslabel>,SYSA,MVS_IMAGE</reslabel>
<restype>MVS_IMAGE</restype>
<reslabelurl>,SYSA,MVS_IMAGE</reslabelurl>
<expandable>YES</expandable>
<icon>rmfmvsim.gif</icon>
<attributes>YES</attributes>
</resource>
…
<resource>
<reslabel>,CF01,COUPLING_FACILITY</reslabel>
<resname>CF01</resname>
<reslabelurl>,CF01,COUPLING_FACILITY</reslabelurl>
<expandable>YES</expandable>
<icon>rmfcf.gif</icon>
<attributes>YES</attributes>
</resource>
</contained>
```

Filter-Instances-List element

The <filter-instances-list> tag lists all possible filter instances with a specific list type for the given resource.

Syntax: Filter-Instances-List



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23
listtype	list type of the requested filter instances	token
instance	element of the instances list	token
Message	error message or warning	See “Message” on page 26

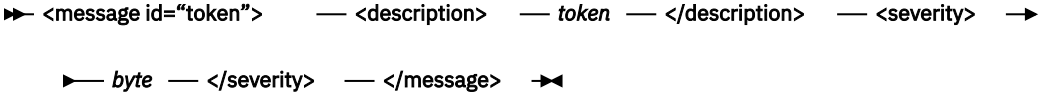
Example XML code:

```
<filter-instances-list>
<resource>
…
</resource>
<listtype>V</listtype>
<instance>SYSA,IMS610,V</instance>
…
<instance>SYSA,IMS710,V</instance>
</filter-instances-list>
```

Message

The <message> tag specifies an error message or warning.

Syntax: Message



Tag/Fragment	Description	Type
id	unique message ID	token
description	message text	token
severity	severity of the message	byte

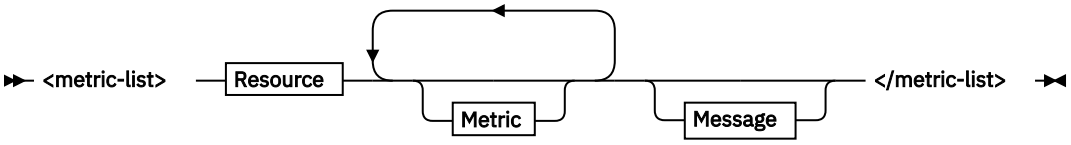
Example XML code:

```
<message id="GPM0626I">
<description>The metric 008D1000 is not defined for resource type SYSPLEX …
</description>
<severity>4</severity>
</message>
```

Metric-List element

The <metric-list> tag lists all available metrics for a given resource.

Syntax: Metric-List



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23
Metric	list of metrics for this resource	See “Metric” on page 27
Message	error message or warning	See “Message” on page 26

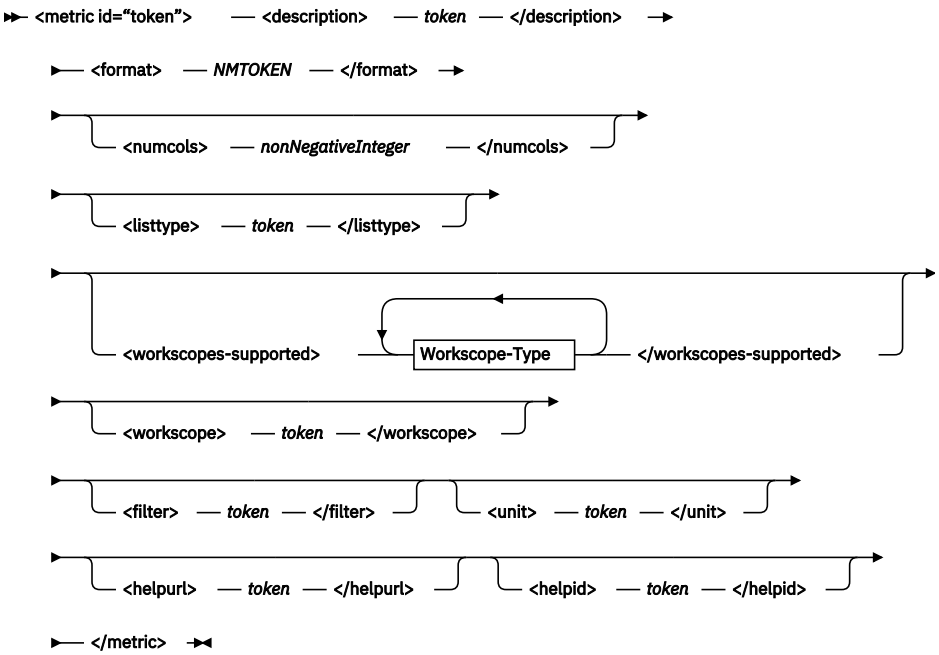
Example XML code:

```
<metric-list>
<resource>
…
</resource>
<metric id="8D2060">
…
</metric>
…
<metric id="8D1FF0">
…
</metric>
</metric-list>
```

Metric

The <metric> tag describes a metric in detail.

Syntax: Metric



Workscope-Type

Tag/Fragment	Description	Type
id	unique metric ID	token
description	description of a metric	token
format	type of counter used for this metric; value is in {single, list, report}	NMTOKEN
numcols	number of columns in report, if metric is a report	nonNegative Integer
listtype	list type of the resource	token
workscopes-supported	valid workscopes for this metric	n/a
workscope-type	valid workscope for this metric; value in {G,W,S,P,R,J}	NMTOKEN
workscope	description of the workscope (see also "The workscope parameter" on page 9)	token
filter	filter argument	token
unit	unit of the values returned by this metric; valid values in {count, index, megabytes, micro-, milliseconds , percent, rate per hour, rate per second, seconds, undefined}	token
helpurl	URL of help document	token
helpid	ID of corresponding paragraph in help document	token

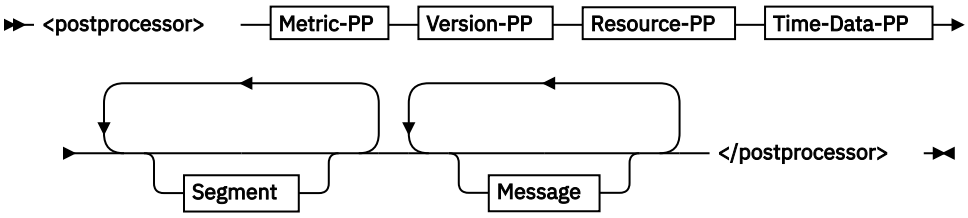
Example XML code:

```
<metric id="8D2060">
<description>% processor utilization</description>
<format>single</format>
<listtype> </listtype>
<workscopes-supported>
<workscope-type>G</workscope-type>
</workscopes-supported>
<unit>percent</unit>
<helpurl>/gpm/include/metrics.html</helpurl>
<helpid>5050</helpid>
</metric>
```

Postprocessor element

The <postprocessor> tag encloses one Postprocessor report denoted by the Metric-PP specification.

Syntax: Postprocessor



Tag/Fragment	Description	Type
Metric-PP	name of a Postprocessor report	See "Metric-PP" on page 29

Tag/Fragment	Description	Type
Version-PP	version information for Postprocessor reports	See "Version-PP" on page 30
Resource-PP	description of the reported resource	See "Resource-PP" on page 30
Time-Data-PP	time information for the report interval	See "Time-Data-PP" on page 30
Segment	named Postprocessor report segment containing the report data	See "Segment" on page 31
Message	error message or warning	See "Message" on page 26

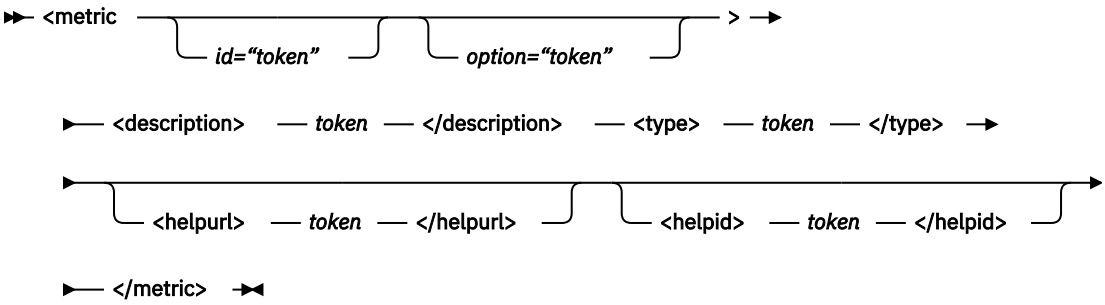
Example XML code:

```
<postprocessor>
<metric id="CPU">…</metric>
<version>…</version>
<resource>…</resource>
<time-data>…</time-data>
<segment><name>CPU ACTIVITY</name>…</segment>
<segment><name>PARTITION DATA REPORT</name>…</segment>
<segment><name>LPAR CLUSTER REPORT</name>…</segment>
<segment><name>GROUP CAPACITY REPORT</name>…</segment>
</postprocessor>
```

Metric-PP

The <metric> tag within the <postprocessor> tag contains the name of a Postprocessor report.

Syntax: Metric-PP



Tag/Fragment	Description	Type
id	name of the Postprocessor report	token
option	suboptions specified for the Postprocessor reports	token
description	descriptive title of the report	token
type	type of the Postprocessor report; can be either Interval, Duration, or Overview	token
helpurl	URL of help document	token
helpid	ID of corresponding paragraph in help document	token

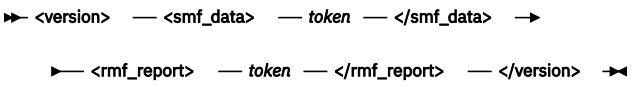
Example XML code:

```
<metric id="CPU">
<description>CPU Activity Report</description>
<type>Interval</type>
</metric>
```

Version-PP

The <version> tag within the <postprocessor> tag contains version information for Postprocessor reports.

Syntax: Version



Tag/Fragment	Description	Type
smf_data	version of the operating system that captured the SMF data	token
rmf_report	version of the RMF Postprocessor	token

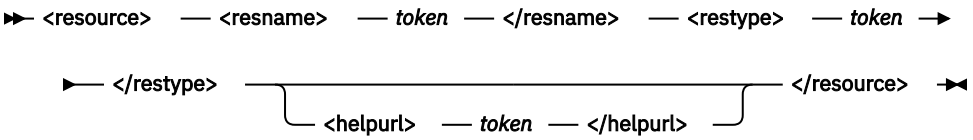
Example XML code:

```
<version>
<smf-data>z/OS V1R12</smf-data>
<rmf-report>V1R12 RMF</rmf-report>
</version>
```

Resource-PP

A <resource> tag within the <postprocessor> tag specifies the reported resource, for example, a system.

Syntax: Resource-PP



Tag/Fragment	Description	Type
resname	resource name	token
restype	resource type	token

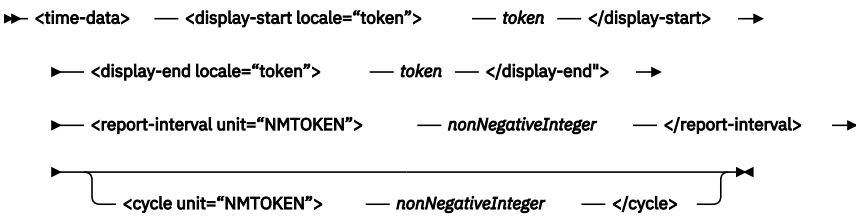
Example XML code:

```
<resource>
<resname>SYSE</resname>
<restype>SYSTEM ID</restype>
</resource>
```

Time-Data-PP

A <time-data> tag within the <postprocessor> tag provides information on the time intervals used in a report.

Syntax: Time-Data-PP



Tag/Fragment	Description	Type
display-start	local start date and time of the reporting interval in displayable format `mm/dd/yyyy-hh:mm:ss`	token
locale	locale for which displayable format is shown	token
display-end	local end date and time of the reporting interval in displayable format `mm/dd/yyyy-hh:mm:ss`	token
report-interval	length of the measurement interval (reporting time)	nonNegative Integer
cycle	cycle length of data sampling	nonNegative Integer
unit	unit of time; valid values in {hours, minutes, seconds, milliseconds, microseconds}	NMTOKEN

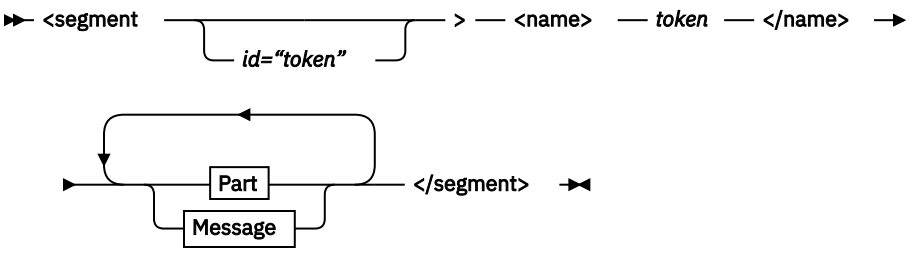
Example XML code:

```
<time-data>
<display-start locale="en-us">03/26/2010-11.15.00</display-start>
<display-end locale="en-us">03/26/2010-11.30.00</display-end>
<report-interval unit="minutes">14:59:999</report-interval>
<cycle unit="milliseconds">1000</cycle>
</time-data>
```

Segment

A tag contains one report section (which can consist of multiple parts, in XML terms) of a Postprocessor report.

Syntax: Segment


<segment
id="token"
>
<name>
token
</name>
Part
Message

Tag/Fragment	Description	Type
id	unique segment ID	token
name	name of a report segment	token
Part	part of a report segment which can contain any number of variable-name/value lists or data tables	See “Part” on page 32

Tag/Fragment	Description	Type
Message	error message or warning	See “Message” on page 26

Example XML code:

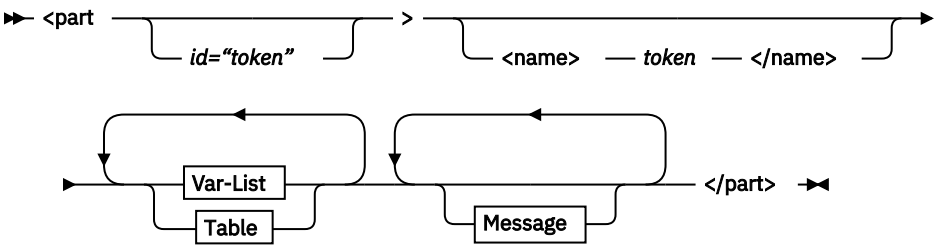
```
<segment><name>CPU ACTIVITY</name>
<part>…</part>
<part>…</part>
…
</segment>
```

See another code example in the description of the <table> tag in “Table” on page 33.

Part

A <part> tag encloses a part of a report segment. Such a part can either contain one or more variable lists (specified by the ***Var-List*** fragment) or one or more data tables (specified by the ***Table*** fragment).

Syntax: Part



Tag/Fragment	Description	Type
id	unique part ID	token
name	name of a part within a report segment	token
Var-List	a list of variable-name/value pairs	See “Var-List” on page 32
Table	a data table within a report part	See “Table” on page 33
Message	error message or warning	See “Message” on page 26

Example XML code:

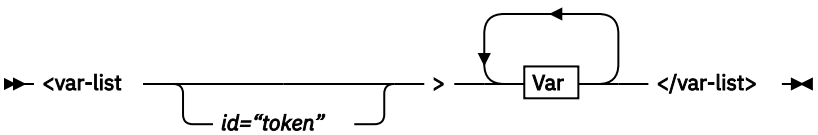
For a <part> code example, see the description of the <table> tag (“Table” on page 33).

Var-List

A <var-list> tag contains a list of variable-name/value pairs, for example:

```
CPU 2817,   Model 715,   H/W Model M15
```

Syntax: Var-List



Tag/Fragment	Description	Type
Var	a variable-name/value pair	See “Var” on page 33

Example XML code:

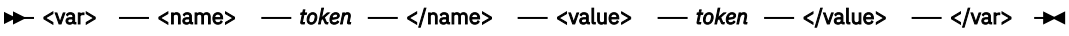
For a <var-list> code example, see the description of the <var> tag (information unit “Var” on page 33).

Var

A <var> tag contains a variable-name/value pair, for example:

```
CPU 2817
```

Syntax: Var



Tag/Fragment	Description	Type
name	name of a variable	token
value	value of a variable	token

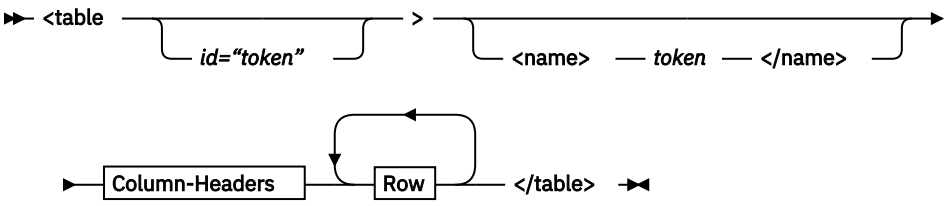
Example XML code:

```
<segment id="1"><name>CPU Activity</name>
<part id="2">
<var-list id="3">
<var><name>CPU</name><value>2817</value></var>
<var><name>Model</name><value>715</value></var>
<var><<name>H/W Model</name><value>M15</value></var>
…
</var-list>
<table id="4">
…
</table>
</part>
<part id="5">
…
</part>
…
</segment>
```

Table

A <table> tag contains a data table, consisting of the column headers and at least one row.

Syntax: Table



Tag/Fragment	Description	Type
id	unique table ID	token
name	name of a table	token
Column-Headers	unique names of the columns in the report table	See “Column-Headers” on page 38

Tag/Fragment	Description	Type
Row	performance data	See “Row” on page 37

Example XML code:

```
<segment id="3"><name>Details for all Channels</name>
<part id="4">
<table id="5">
<column-headers>
<col type="T">Channel Path ID</col>
<col type="T">Type</col>
<col type="T">Generation</col>
<col type="T">Shared</col>
<col type="T">Status</col>
<col type="N">Util% (LPAR)</col>
…
</column-headers>
<row refno="1">
<col>0D</col>
<col>OSD</col>
<col/>
<col>Yes</col>
<col>Online</col>
<col>0.24</col>
…
</row>
<row refno="2">…</row>
</table></part></segment>
```

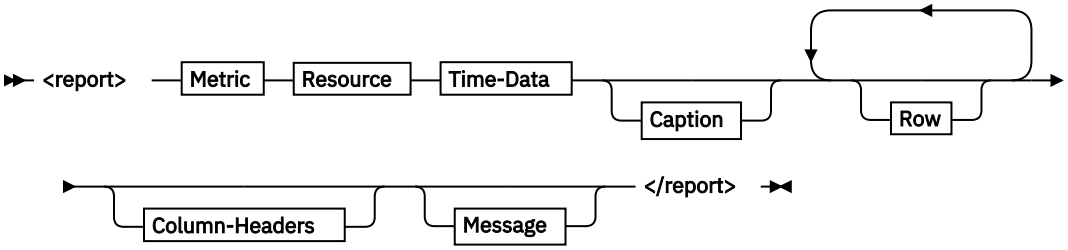
Resulting browser display:

```
Details for all Channels

Channel                                    Util%
Path ID  Type  Generation  Shared Status  (LPAR) …

0D       OSD               Yes    Online  0.24      …
30       FC_S  9           Yes    Online  0.04      …
```

Report element

The <report> tag encloses performance data for a specific resource or metric.

Syntax: Report



Tag/Fragment	Description	Type
Metric	metric used for the report	See “Metric” on page 27
Resource	description of a resource	See “Resource” on page 23
Time-Data	time and interval information for the report	See “Time-Data” on page 35
Caption	additional (sub-) headings for the report	See “Caption” on page 36
Row	performance data	See “Row” on page 37

Tag/Fragment	Description	Type
Column-Headers	unique names of the columns in the report table	See “Column-Headers” on page 38
Message	error message or warning	See “Message” on page 26

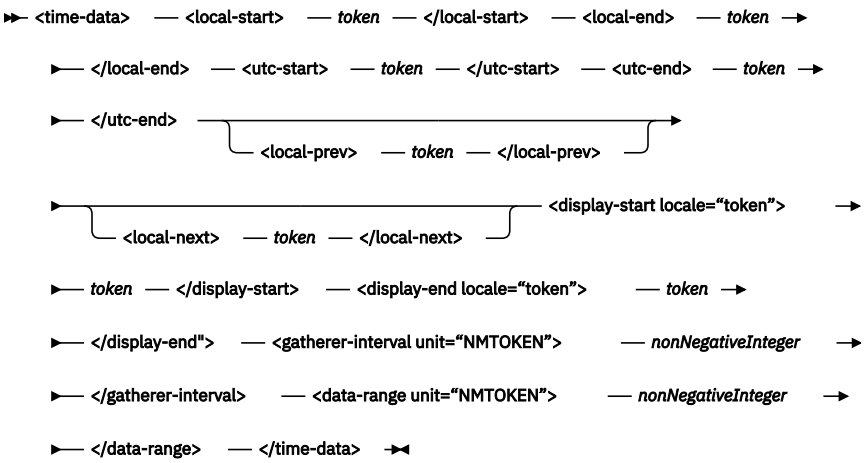
Example XML code:

```
<report>
<metric id="CPC">
…
</metric>
<resource>
…
</resource>
<time-data>
…
</time-data>
<caption>
…
</caption>
<row refno="1">
…
</row>
…
<row refno="15">
…
</row>
<column-headers>
…
</column-headers>
</report>
```

Time-Data

The <time-data> tag provides information on the time intervals used in a report.

Syntax: Time-Data



Tag/Fragment	Description	Type
local-start	local start time of data range in format yyyymmddhhmmss (a sequence of 14 digits)	token
local-end	local end time of data range in format yyyymmddhhmmss	token
utc-start	start of data range, specified as coordinated universal time in the format yyyymmddhhmmss	token

Tag/Fragment	Description	Type
utc-end	end of range, specified as coordinated universal time in the format yyyymmddhhmmss	token
local-prev	local timestamp of previous sample in format yyyymmddhhmmss	token
local-next	local timestamp of next sample in format yyyymmddhhmmss	token
display-start	local start time of data range in displayable format mm/dd/yyyy hh:mm:ss	token
locale	locale for which displayable format is shown	token
display-end	local end time of data range in displayable format mm/dd/yyyy hh:mm:ss	token
gatherer-interval	length of gatherer interval	nonNegative Integer
unit	unit of time; valid values in {hours, microseconds, milliseconds, minutes, seconds}	NMTOKEN
data-range	length of data range	nonNegative Integer

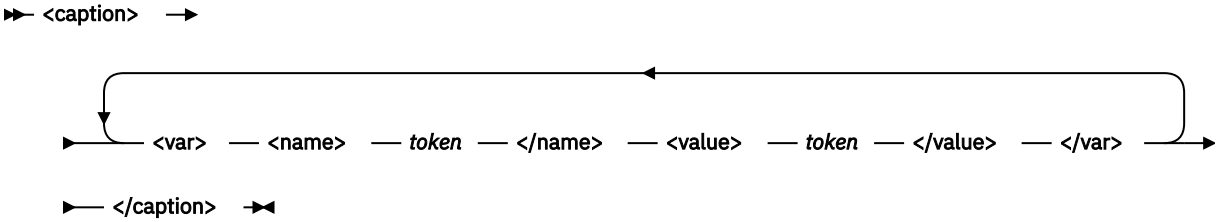
Example XML code:

```
<time-data>
<local-start>20190214180800</local-start>
<local-end>20190214180830</local-end>
<utc-start>20190214170800</utc-start>
<utc-end>20190214170830</utc-end>
<local-prev>20190214180745</local-prev>
<local-next>20190214180845</local-next>
<display-start locale="en-us">02/14/2019 18:08:00</display-start>
<display-end locale="en-us">02/14/2019 18:08:30</display-end>
<gatherer-interval unit="seconds">30</gatherer-interval>
<data-range unit="seconds">30</data-range>
</time-data>
```

Caption

The <caption> tag contains additional headings and summary information for a report.

Syntax: Caption



Tag/Fragment	Description	Type
var	description of variables in additional headings	n/a
name	name of variable	token
value	value of variable	token

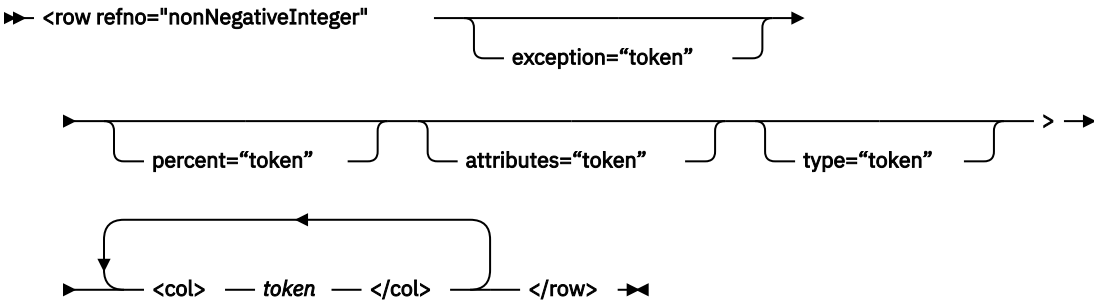
Example XML code:

```
<caption>
<var>
<name>CPCHPNAM</name>
<value>SYSA</value>
</var>
…
<var>
<name>CPCHMOD</name>
<value>2817</value>
</var>
</caption>
```

Row

The <row> tag contains the requested performance data values to be displayed in a Monitor III or Postprocessor report.

Syntax: Row



Tag/Fragment	Description	Type
col	value for cells in a row of a data table	token
refno	unique row reference number	nonNegative Integer
exception	indicates if the value in this row exceeds exception thresholds; valid values in {NONE, WARN, CRIT}	NMTOKEN
percent	percentage of the value compared to the maximum value in the list (only meaningful for list-valued metrics)	token
attributes	denotes whether attributes may be queried for the resource in this row; valid values in {YES, NO}	NMTOKEN
type	is set if the row contains values with a special meaning, for example, SUM denotes a row containing a total of certain previous rows	NMTOKEN

Example XML code of a row in a Postprocessor report:

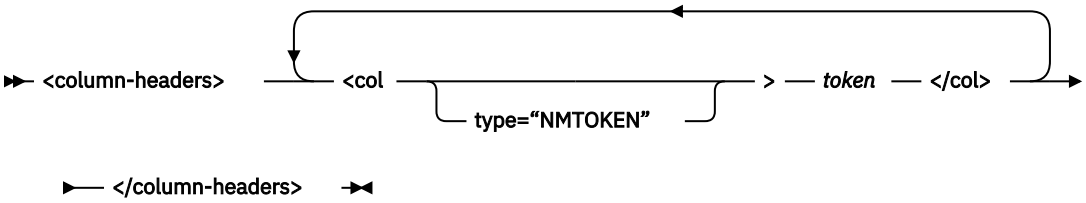
```
<row refno="1" type="SUM">
<col>TOTAL/AVERAGE</col>
<col>CP</col>
<col>3.08</col>
<col>3.05</col>
<col>191.2</col>
<col>7.45</col>
<col>0.58</col>
</row>
```

Column-Headers

The <column-headers> tag contains the names of the columns in the report table:

- For Monitor III data, these names are equal to the ISPF variable names in the Monitor III ISPF reports.
- For Postprocessor data, these names are the field headings in the data tables.

Syntax: Column-Headers



Tag/Fragment	Description	Type
col	unique name of the column	token
type	data type in this column; valid values in {M, N, S, T} for either ***M***ixed alphanumerics, ***N***umbers, ***S***caled numbers, or ***T***ext	NMTOKEN

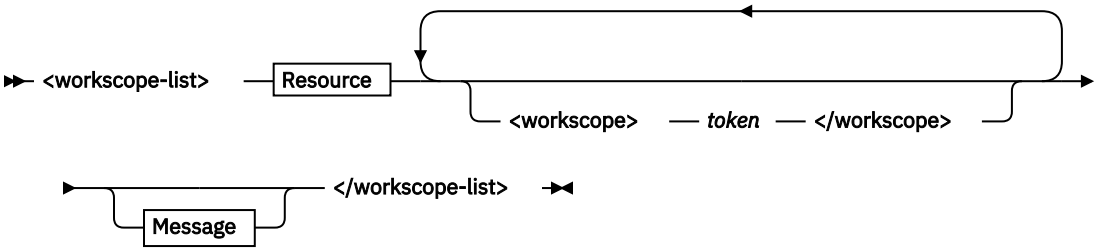
Example XML code:

```
<column-headers>
<col type="T">CPCPPNAM</col>
<col type="N">CPCPDMSU</col>
…
<col type="N">CPCPLTOU</col>
</column-headers>
```

Workscope-List element

The <workscope-list> tag lists the available workscopes for a given resource.

Syntax: Workscope-List



Tag/Fragment	Description	Type
Resource	description of a resource	See “Resource” on page 23
workscope	description of a workscope	token
Message	error message or warning	See “Message” on page 26

Example XML code:

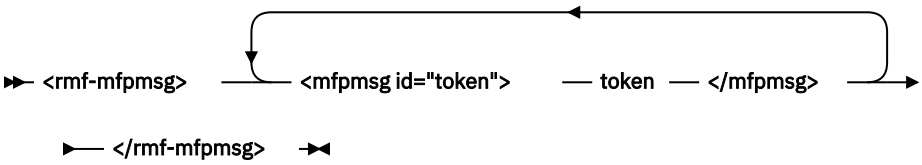
```
<workscope-list>
<resource>
…
</resource>
<workscope>,BATCH,W</workscope>
…
```

```
<workscope>,HOTTSO,S</workscope>
</workscope-list>
```

RMF-MFP-Message element

The <rmf-mfpmsg> tag lists the messages which have been generated by RMF Postprocessor and are provided in the MFPMSGDS SYSOUT DD of the Postprocessor job run by the RMF Distributed Data Server.

Syntax: RMF-MFP-Message



Tag/Fragment	Description	Type
mfpmsg	message text	token
id	Postprocessor message ID	token

Example XML code:

```
<rmf-mfpmsg>
   <mfpmsg id="ERB103I"> PPS: OPTIONS IN EFFECT</mfpmsg>
   <mfpmsg id="ERB103I"> PPS: ETOD(0000,2400) -- DEFAULT</mfpmsg>
   <mfpmsg id="ERB103I"> PPS: NOSUMMARY -- SYSIN</mfpmsg>
   <mfpmsg id="ERB103I"> PPS: REPORTS(CPU) -- SYSIN</mfpmsg>
   <mfpmsg id="ERB103I"> PPS: DATE(01011967,12312066) -- DEFAULT</mfpmsg>
</rmf-mfpmsg>
```

Coding example for requesting and receiving Monitor III performance data

The subsequent Java™ code sample demonstrates a method how to send a HTTP requests to the DDS and how to receive the response into a text file, from where you can extract the required values.

```
import java.io.*;
import java.net.*;
import sun.misc.*;

/*************************************************************
 * A command line program to communicate with the DDS server  *
 *************************************************************/

public class ParseDDS {

public final static String URLPREFIX =
       "http://<ddshost>:<ddsport>/gpm/";
public final static String OUTFILE = "ParseDDS.txt";

PrintWriter writer;

/*************************************************************************
 * Constructor: Builds a performance data request and sends it to the DDS. *
 * Writes results into a local file.                                      *
 *************************************************************************/

public ParseDDS() {
 try {
       String urlstr;
       writer = new PrintWriter(new FileOutputStream(OUTFILE, false));
       for (int i = 0; i < 1; i++) {

       /* Get performance data (Performance index of all service class periods) */

       urlstr = URLPREFIX + "perform/perform.xml?resource=,*,SYSPLEX&id=8D1020";
       getAndWrite(urlstr);

       }
```

```
 }
 catch (Exception e) {
        e.printStackTrace();
 }

 writer.close();

}

/*********************************************************
 * Sends request to server and writes XML data to file  *
 *********************************************************/

public void getAndWrite(String urlstr) throws Exception {
       URL url = new URL(urlstr);
       HttpURLConnection con = (HttpURLConnection) url.openConnection();
       String line;
       con.setDoInput(true);
       con.connect();
       int cl = con.getContentLength();

       /* Write HTTP contents (XML document) to file */

       line = "URL=" + urlstr;
       System.out.println(line);
       writer.println(line);
       if (cl > 0) {
              BufferedReader reader =
                 new BufferedReader(new InputStreamReader(con.getInputStream()));
              int k = 0;
              while ((line = reader.readLine()) != null) {
                      k++;
                      System.out.println(line);
                      writer.println(line);
              }
              System.out.println(
                      "   " + cl + " Bytes XML content received from DDS.");
       }
       System.out.println(" ");
}

/*************************************************************
 *                            main                           *
 *************************************************************/

public static void main(String[] args) {

     System.out.println(
             "ParseDDS started. Output will be written to file: " + OUTFILE);
     ParseDDS myDDs = new ParseDDS();
     System.out.println("ParseDDS ended.");
     System.exit(0);
  }
}
```

Chapter 2. z/OS CIM monitoring

The Common Information Model (CIM) is a standard data model developed by a consortium of major hardware and software vendors (including IBM), called the Distributed Management Task Force (DMTF). It provides a common definition for describing and accessing systems management data in heterogeneous environments. It allows vendors and system programmers to write applications (CIM monitoring clients) that measure system resources in a network with different operating systems and hardware, and to actually manage those systems.

The z/OS base element ***Common Information Model*** (z/OS CIM) implements the ***CIM server***, based on the OpenPegasus open source project. A CIM monitoring client invokes the CIM server, which in turn collects z/OS metrics from the system and returns it to the calling client.

z/OS CIM Monitoring provides access to z/OS resources and monitoring data.

If a CIM client requests the CIM server to obtain z/OS metrics, the CIM server invokes the appropriate ***z/OS RMF monitoring provider*** which retrieves these metrics associated to z/OS system resources. The ***z/OS RMF monitoring providers*** use RMF Monitor III performance data.

For more information on the z/OS CIM element refer to *z/OS Common Information Model User's Guide*.

z/OS RMF CIM monitoring

z/OS RMF CIM monitoring for z/OS is available only if RMF is running and the RMF DDS is started. How to provide these prerequisites is described in Starting and stopping RMF, and in "Setting up the Distributed Data Server" and "Starting the Distributed Data Server" in *z/OS RMF Reporter User's Guide*. How to set the required environment variables for the CIM server is described in *z/OS Common Information Model User's Guide*.

z/OS RMF CIM monitoring uses metrics that are associated with resource classes implemented by the z/OS RMF monitoring providers described in *z/OS Common Information Model User's Guide*.

The following CIM classes have been implemented as IBM supplied providers according to the DMTF dynamic metrics model. You can find more information about this data model in the *CIM Metrics White Paper (DSP0141)*, which is available at the DMTF website (www.dmtf.org).

- ***IBMzOS_BaseMetricDefinition***: representation of metric definition (for example, metric name and ID); a subclass of ***CIM_BaseMetricDefinition***
- ***IBMzOS_BaseMetricValue***: representation of a metric value (with value, time stamp and duration); a subclass of ***CIM_BaseMetricValue***
- ***IBMzOS_MetricDefForME***: association between a managed element (resource) and metric definition instances; a subclass of ***CIM_MetricDefForME***.
- ***IBMzOS_MetricForME***: association between a managed element (resource) and metric value instances; a subclass of ***CIM_MetricForME***.
- IBMzOS_BaseMetricInstance: association between metric definition and metric value instances; a subclass of ***CIM_MetricInstance***.

All instances of IBMzOS_BaseMetricValue will return volatile data. Historical data is not supported with this release.

For a list of the metrics supported in z/OS, see "z/OS metrics" on page 48. Also see the CIM class and instance MOF files. The MOF file *IBMzOS_Monitoring.mof* can be found in `/usr/lpp/wbem/provider/schemas/os_monitoring`.

The following diagram illustrates the relationship between the IBM extension classes, and the CIM base classes they extend:

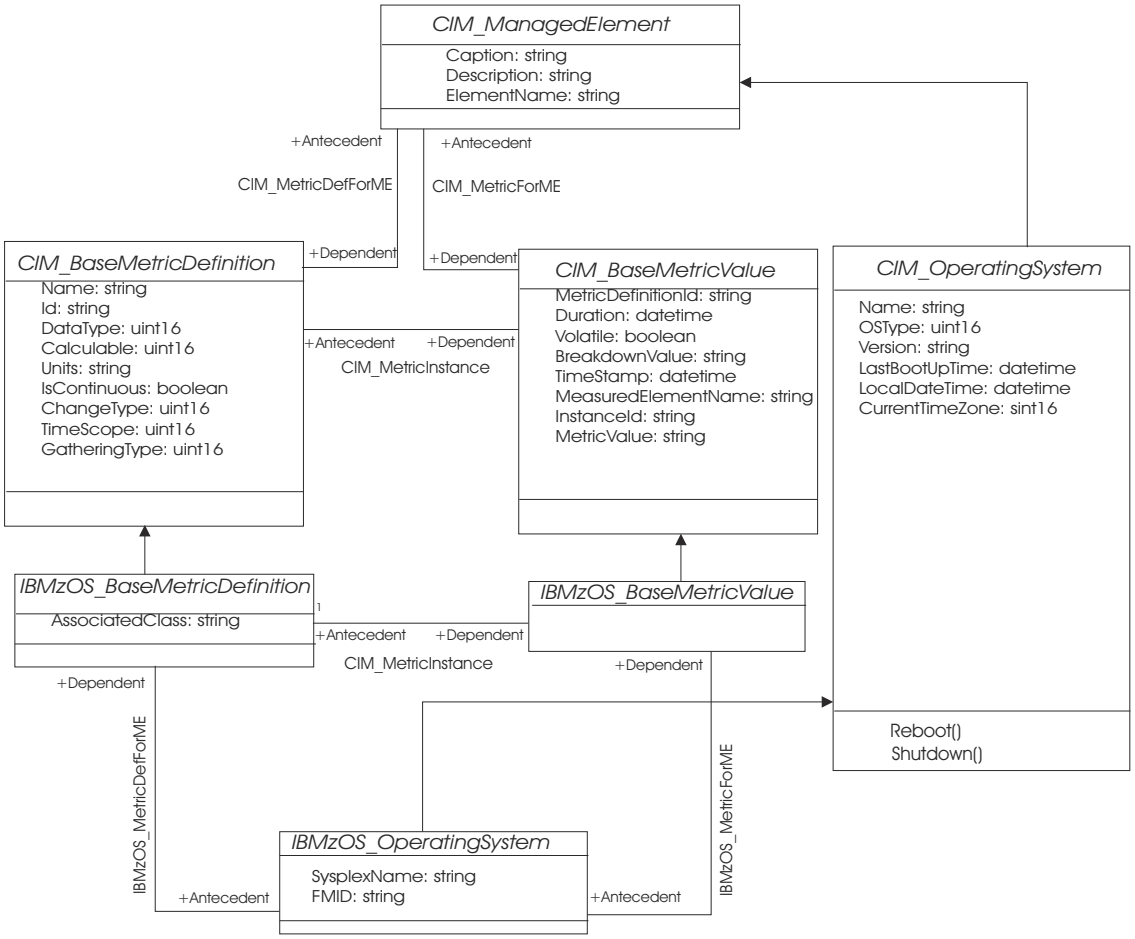


Figure 6. z/OS RMF implementation of the DMTF dynamic metrics model

Class ***IBMzOS_OperatingSystem*** is implemented by the *z/OS OS Management Providers* (part of base element z/OS CIM). It is just an example of a resource which can be inherited from ***CIM_OperatingSystem***. Class ***CIM_OperatingSystem*** can be implemented with the same attributes on any operating system that exploits the CIM model. ***IBMzOS_OperatingSystem*** inherits all properties from this base class and implements further z/OS-specific attributes, like `SysplexName` and `FMID`. BaseMetricValue and BaseMetricDefinition instances can be associated to it.

CIM_BaseMetricDefinition:

An instance of this class represents how a metric is defined. The associated class CIM_BaseMetricValue holds the metric value. CIM_BaseMetricDefinition provides a way to introduce a new metric definition at run time and capture its instance values in a separate class.

CIM_BaseMetricValue:

Each instance of this class represents a metric value.

CIM_MetricDefForME/IBMzOS_MetricDefForME:

Traversing this association from a resource returns the set of all metric definitions for the given resource. **Usage example:** If you want to know what metric definitions are available for the ***CIM_Process*** resource, you can use this association. This association returns static data, which does not change without applying service to z/OS CIM. Therefore call it once in your application at startup time to figure out what is available. You do not need to traverse it several times.

CIM_MetricForME/IBMzOS_MetricForME:

This association links a measured element (resource instance) to all metric instances available for it. **Usage example:** Traversing this association starting from an ***IBMzOS_OperatingSystem*** instance, returns

all associated ***IBMzOS_BaseMetricValue*** instances. In other words, this association returns metrics for the z/OS image on which the CIM server is running.

CIM_MetricInstance/IBMzOS_MetricInstance:

Traversing this association gives you all metric instances available for a given ***CIM_BaseMetricDefinition***. **Usage example:** In order to get the metric values for the ResidentSetSize (working set size) of all process instances (z/OS address spaces) you can start at the metric definition instance of the ResidentSetSize instance. Traversing the ***CIM_MetricInstance*** association returns all instances of the ResidentSetSize metric for all address spaces in your system.

z/OS resource classes based on RMF

This section documents those CIM resource classes available for z/OS that are implemented based on RMF data.

Note: All described classes are only available if RMF is up and running on the system where the monitored resources are located.

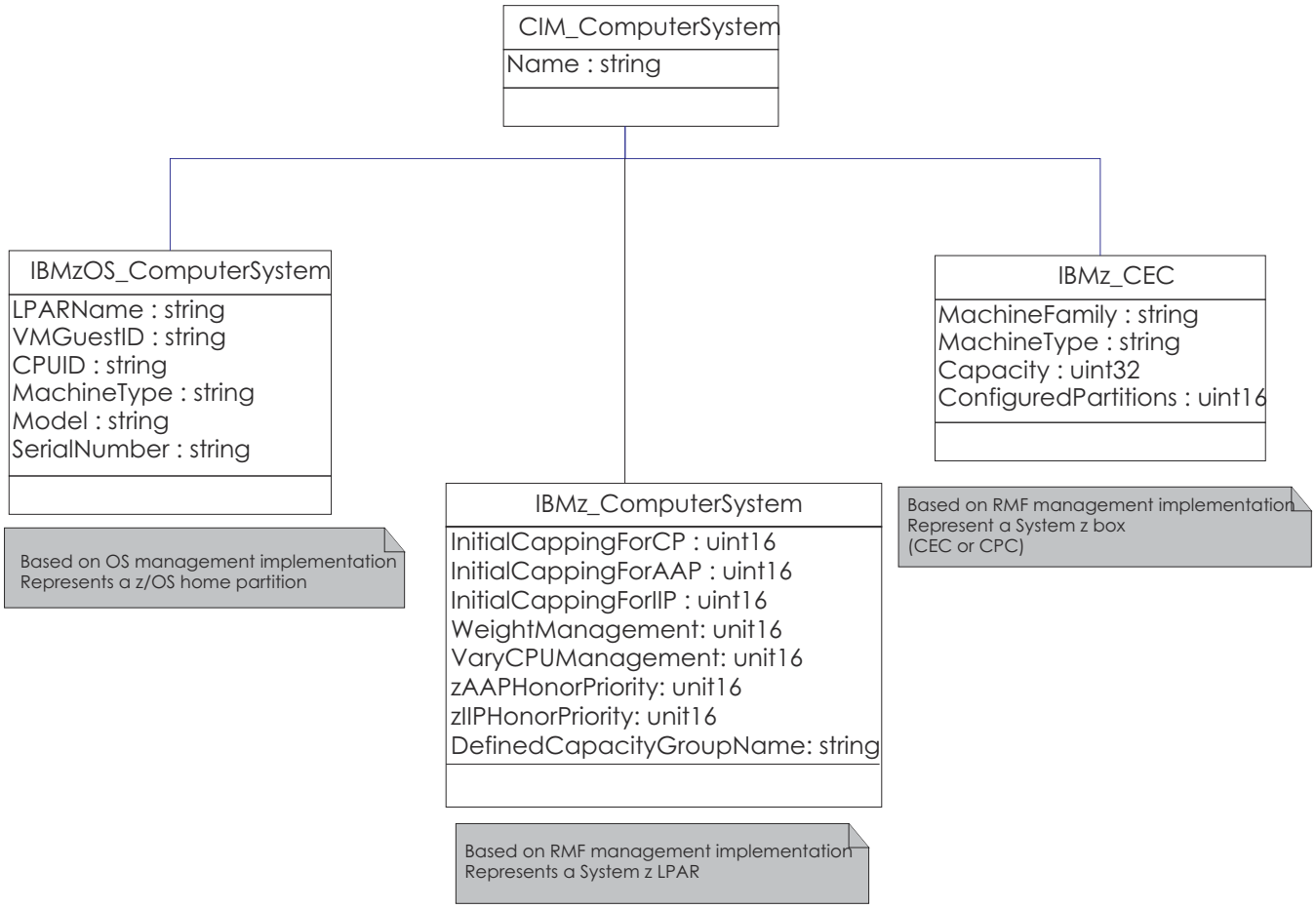


Figure 7. z/OS resource classes implemented by RMF

IBMz_CEC

This resource represents a IBM Z box including processors, memory, I/O cages and so on. Note that CEC (central electronics complex) is a commonly used synonym for CPC (central processing complex). IBMz_CEC contains the following attributes:

Property	Description
string Name (key)	CEC serial number

Property	Description
string CreationClassName (key)	Value “IBMz_CEC”
string Machine Family	Processor type, for example, “2094”
string Machine Type	Software model of the processor, for example, “716”
uint32 Capacity	System capacity in MSU/hour
uint16 ConfiguredPartitions	Number of configured partitions in the CEC

IBMz_ComputerSystem

This class represents a logical partition (LPAR) and contains the following attributes:

Property	Description
string Name (key)	LPAR name
string CreationClassName (key)	Value “IBMz_ComputerSystem”
string CECName (key)	Name of the CEC this LPAR exists on
uint16 InitialCappingForCP	Information about initial capping for this partition (logical processor type CP): **0** unknown **1** other **2** enabled **3** disabled
uint16 InitialCappingForAAP	Information about initial capping for this partition (logical processor type zAAP): **0** unknown **1** other **2** enabled **3** disabled
uint16 InitialCappingForIIP	Information about initial capping for this partition (logical processor type zIIP): **0** unknown **1** other **2** enabled **3** disabled

Property	Description
uint16 WeightManagement	Information about z/OS IRD LPAR weight management: **0** unknown **1** other **2** enabled **3** disabled
uint16 VaryCPUManagement	Information about z/OS IRD vary CPU management: **0** unknown **1** other **2** enabled **3** disabled
uint16 zAAPHonorPriority	Information about zAAP honor priority: **0** unknown **1** other **2** enabled **3** disabled
uint16 zIIPHonorPriority	Information about zIIP honor priority: **0** unknown **1** other **2** enabled **3** disabled
string DefinedCapacityGroupName	Name of the capacity group to which the partition belongs, if it is managed towards a common group capacity limit.
real32 HWcapLimitCP	The defined hardware capping limit in numbers of CPUs for general purpose processors for the partition.
real32 HWcapLimitAAP	The defined hardware capping limit in numbers of CPUs for specialty processors of type AAP for the partition.
real32 HWcapLimitIIP	The defined hardware capping limit in numbers of CPUs for specialty processors of type IIP for the partition.

Property	Description
string DefinedHWGroupName	The name of the hardware group which the partition belongs to, if it is managed towards a common hardware group capping limit.
real32 HWGroupCapLimitCP	The defined hardware group capping limit in numbers of CPUs for general purpose processors for the hardware group which the partition belongs to.
real32 HWGroupCapLimitIIP	The defined hardware group capping limit in numbers of CPUs for specialty processors of type IIP for the hardware group which the partition belongs to.
uint16 AbsMSUCap	Information about absolute MSU capping for this partition. **0** unknown **1** other **2** enabled **3** disabled
uint16 zIIPBoost	zIIP boost information **0** zIIP boost was inactive **1** zIIP boost was active
uint16 SpeedBoost	Speed boost information **0** Speed boost was inactive **1** Speed boost was active

IBMzOS_Channel

This resource represents a channel path in the computer system, based on RMF information. IBMzOS_Channel contains the following attributes:

Property	Description
string DeviceID (key)	Channel path ID (CHPID)
string SystemName (key)	z/OS MVS system name
string ChannelPathType	Type of channel path

IBMzOS_WLMServiceDefinition

This class represents the z/OS WLM policy. It is a subclass from ***CIM_ManagedElement*** and contains the following attributes:

Property	Description
string Name (key)	Name of the WLM service definition
string PolicyName	Name of the active WLM service policy

Property	Description
datetime PolicyActivationTime	Date and time the WLM service policy has been activated

IBMzOS_WLMServiceClassPeriod

This class provides basic properties of a service class period defined for a WLM service class and contains the following attributes:

Property	Description
string ServiceClassName	Name of the WLM service class to which this service class period belongs
string PeriodNumber	Service class period in which the service class is currently running
uint16 ImportanceLevel	Importance level ranging form 1 to 5 where 1 is the most important level. If the property value cannot be determined, for example, for inactive service classes periods or for service class periods with importance ‘discretionary’, NULL is returned.

z/OS metrics

This chapter describes the CIM metrics, as they are supported by z/OS RMF, with z/OS specific details.

In the following, this chapter contains a subsection for each resource class which lists the metrics available for these classes.

IBMzOS_LogicalDisk

This class represents a logical disk in the system. The following metrics are associated to IBMzOS_LogicalDisk:

Property type	Property value	Property type	Property value
ActiveTimePercentage			
Description: Percentage of time the disk unit was actively processing requests, calculated as the sum of connect, disconnect and pending time. This metric cannot exceed 100%, and it does not state how many requests were active at any given time.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% active time	***DDS metric ID***	8D0010
Associated class	IBMzOS_LogicalDisk		
AvailableSpace			
Description: Free capacity on the disk drive.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	freespace	***DDS metric ID***	8D3090
Associated class	IBMzOS_LogicalDisk		
AverageDeviceUtilization			
Description: Average device utilization (not normalized to 100% for parallel I/O activity). This metric can be above 100% if the device is executing multiple I/Os in parallel. A busy time percentage of 200% means that on average, the device was executing two I/Os in parallel. This metric does not tell you whether the busy time of the measured device could be even higher, because this is architecture dependent.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	i/o activity rate and response time	***DDS metric ID***	based on 8D0E90 and 8D10F0
Associated class	IBMzOS_LogicalDisk		
Capacity			
Description: Capacity of the disk drive.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	capacity	***DDS metric ID***	8D2FF0
Associated class	IBMzOS_LogicalDisk		
FastWriteRate			
Description: Rate of write operations executed as fast writes.			
Datatype	real32	***Units***	1/s
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	cache DFW hits (all systems)	***DDS metric ID***	8D21D0
Associated class	IBMzOS_LogicalDisk		
IOIntensity			
Description: I/O utilization indicator: IOIntensity = ResponseTime * IORate			
Datatype	real32	***Units***	milliseconds per second
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	i/o intensity	***DDS metric ID***	8D1290
Associated class	IBMzOS_LogicalDisk		
QueueDepth			
Description: Average number of I/O requests currently in queue (OS view).			
Datatype	real32	***Units***	s/s = I/O request rate [1/s] * average response time [ms] / 1000
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	i/o activity rate and response time	***DDS metric ID***	based on 8D0E90 and 8D10F0
Associated class	IBMzOS_LogicalDisk		
ReadCacheHitRate			
Description: Rate of read requests that did not need access to disk drives because data was available in cache.			
Datatype	real32	***Units***	1/s
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	cache read hits (all systems)	***DDS metric ID***	8D2280
Associated class	IBMzOS_LogicalDisk		
RequestRate			
Description: Number of I/O requests per second for the associated device.			
Datatype	real32	***Units***	1 / s (events per second)
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	i/o activity rate	***DDS metric ID***	8D0E90
Associated class	IBMzOS_LogicalDisk		
ResponseTime			
Description: ResponseTime associated to a logical disk drive. The average response time (in milliseconds) that the device required to complete an I/O request. For comparison of average response times on different platforms, please keep in mind that this metric may be reported by the disk device itself or it may be computed by the operating system, beginning at the point in time when an application issues a disk related command and ending at the point in time when the data is returned. In this case, the queue times, network times and other components of response time are included in the ResponseTime metric. In z/OS, this metric represents the operating system view of the disk response time.			
Datatype	real32	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	response time	***DDS metric ID***	8D10F0
Associated class	IBMzOS_LogicalDisk		
WaitTime			

Property type	Property value	Property type	Property value
Description: WaitTime associated to a logical disk drive. This metric comprises an estimation of the delay components of ***ResponseTime*** (in milliseconds), or ***AverageResponseTime*** minus service time, and the time spent in queues.			
Datatype	real32	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% connect time and response time	***DDS metric ID***	based on 8D00B0 and 8D10F0
Associated class	IBMzOS_LogicalDisk		

IBMz_CEC

This class contains basic properties of a CEC box of a IBM Z.

Note: Not only that CEC is instrumented on which RMF is running, but all CECs of the sysplex. If z/OS is running as a guest under z/VM, **IBMz_CEC** instances and associated metrics are not available.

Property type	Property value	Property type	Property value
LPARWeightForAAP			
Description: LPAR weight for processor type zAAP.			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (AAP)	***DDS metric ID***	8D3F30
Associated class	IBMz_CEC		
LPARWeightForCP			
Description: LPAR weight for standard processor.			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (CP)	***DDS metric ID***	8D3F60
Associated class	IBMz_CEC		
LPARWeightForICF			
Description: LPAR weight for processor type ICF.			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (ICF)	***DDS metric ID***	8D3F90
Associated class	IBMz_CEC		
LPARWeightForIFL			
Description: LPAR weight for processor type IFL.			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	LPAR weight (IFL)	***DDS metric ID***	8D3FE0
Associated class	IBMz_CEC		
LPARWeightForIIP			
Description: LPAR weight for processor type zIIP.			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (IIP)	***DDS metric ID***	8D4010
Associated class	IBMz_CEC		
NumberOfDedicatedCPs			
Description: Number of dedicated standard processors.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfDefinedAAPs			
Description: Number of defined processors of type zAAP.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfDefinedCPs			
Description: Number of defined standard processors.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none

Property type	Property value	Property type	Property value
Associated class	IBMz_CEC		
NumberOfDefinedICFs			
Description: Number of defined processors of type ICF.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfDefinedIFLs			
Description: Number of defined processors of type IFL.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfDefinedIIPs			
Description: Number of defined processors of type zIIP.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfSharedAAPs			
Description: Number of shared processors of type zAAP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		

Property type	Property value	Property type	Property value
NumberOfSharedCPs			
Description: Number of shared standard processors.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
NumberOfSharedIIPs			
Description: Number of zIIPs in zIIP shared pool (shared physicals).			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMz_CEC		
SumOfAAPsAcrossLPARs			
Description: Sum of shared logical zAAP processors or cores across all LPARs in CEC.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# logical processors shared (AAP)	***DDS metric ID***	8D3B70
Associated class	IBMz_CEC		
SumOfCPsAcrossLPARs			
Description: Sum of CPs across all LPARs in CEC (shared logicals).			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# logical processors shared (CP)	***DDS metric ID***	8D3BA0
Associated class	IBMz_CEC		
SumOfIIPsAcrossLPARs			
Description: Sum of shared logical zIIP processors or cores across all LPARs in CEC.			

Property type	Property value	Property type	Property value
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# logical processors shared (IIP)	***DDS metric ID***	8D3C50
Associated class	IBMz_CEC		
SumOfOnlineAAPsAcrossLPARs			
Description: Sum of online logical zAAP processors or cores across all LPARs in CEC.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (AAP)	***DDS metric ID***	8D3C80
Associated class	IBMz_CEC		
SumOfOnlineCPsAcrossLPARs			
Description: Sum of online CPs across all LPARs in CEC (shared logicals).			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (CP)	***DDS metric ID***	8D3CA0
Associated class	IBMz_CEC		
SumOfOnlineIIPsAcrossLPARs			
Description: Sum of online logical zIIP processors or cores across all LPARs in CEC.			
Datatype	uint32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (IIP)	***DDS metric ID***	8D3D50
Associated class	IBMz_CEC		
TotalAAPTimePercentage			
Description: Total physical zAAP utilization percentage (CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)

Property type	Property value	Property type	Property value
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (AAP)	***DDS metric ID***	8D3300
Associated class	IBMz_CEC		
TotalCPTimePercentage			
Description: Total physical CP utilization percentage (CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (CP)	***DDS metric ID***	8D2540
Associated class	IBMz_CEC		
TotalIIPTimePercentage			
Description: Total physical zIIP utilization percentage (CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (IIP)	***DDS metric ID***	8D1C70
Associated class	IBMz_CEC		
TotalSharedAAPTimePercentage			
Description: Total physical ZAAP utilization percentage (shared zAAP, CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (AAP)	***DDS metric ID***	8D3910
Associated class	IBMz_CEC		
TotalSharedCPTimePercentage			
Description: Total physical CP utilization percentage (shared CP, CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	% of total physical utilization (shared CP)	***DDS metric ID***	8D3920
Associated class	IBMz_CEC		
TotalSharedICFTimePercentage			
Description: Total physical ICF utilization percentage (shared ICF, CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (shared ICF)	***DDS metric ID***	8D3930
Associated class	IBMz_CEC		
TotalSharedIFLTimePercentage			
Description: Total physical IFL utilization percentage (shared IFL, CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (shared IFL)	***DDS metric ID***	8D3940
Associated class	IBMz_CEC		
TotalSharedIIPTimePercentage			
Description: Total physical zIIP utilization percentage (shared zIIP, CEC level).			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of total physical utilization (shared IIP)	***DDS metric ID***	8D3950
Associated class	IBMz_CEC		

IBMz_ComputerSystem

Instances of the ***IBMz_ComputerSystem*** class represent IBM Z logical partitions (LPARs). If z/OS is running under z/VM, ***IBMz_ComputerSystem*** instances and associated metrics are not available.

Property type	Property value	Property type	Property value
LPARWeightForAAP			
Description: LPAR weight (processor type zAAP).			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (AAP)	***DDS metric ID***	8D3F20
Associated class	IBMz_ComputerSystem		
LPARWeightForCP			
Description: LPAR weight (processor type CP).			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (CP)	***DDS metric ID***	8D3F50
Associated class	IBMz_ComputerSystem		
LPARWeightForICF			
Description: LPAR weight (processor type ICF).			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (ICF)	***DDS metric ID***	8D3F80
Associated class	IBMz_ComputerSystem		
LPARWeightForIIP			
Description: LPAR weight (processor type zIIP).			
Datatype	uint32	***Units***	weight
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	LPAR weight (IIP)	***DDS metric ID***	8D4000

Property type	Property value	Property type	Property value
Associated class	IBMz_ComputerSystem		
MTCapFactorCP			
Description: MT capacity factor for CP			
Datatype	real64	***Units***	MT capacity factor
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT capacity factor for CP	***DDS metric ID***	8D4B10
Associated class	IBMz_ComputerSystem		
MTCapFactorIIP			
Description: MT capacity factor for IIP			
Datatype	real64	***Units***	MT capacity factor
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT capacity factor for IIP	***DDS metric ID***	8D4B30
Associated class	IBMz_ComputerSystem		
MTMaxCapFactorCP			
Description: MT maximum capacity factor for CP			
Datatype	real64	***Units***	MT capacity factor
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT maximum capacity factor for CP	***DDS metric ID***	8D4B70
Associated class	IBMz_ComputerSystem		
MTMaxCapFactorIIP			
Description: MT maximum capacity factor for IIP			
Datatype	real64	***Units***	MT capacity factor
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT maximum capacity factor for IIP	***DDS metric ID***	8D4B90
Associated class	IBMz_ComputerSystem		

Property type	Property value	Property type	Property value
MTModeCP			
Description: MT mode for processor type CP			
Datatype	uint16	***Units***	MT mode
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT mode for CP	***DDS metric ID***	8D4BD0
Associated class	IBMz_ComputerSystem		
MTModeIIP			
Description: MT mode for processor type IIP			
Datatype	uint16	***Units***	MT mode
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	MT mode for IIP	***DDS metric ID***	8D4BF0
Associated class	IBMz_ComputerSystem		
NumberOfDedicatedAAPs			
Description: Number of dedicated processors or cores of type zAAP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors dedicated (AAP)	***DDS metric ID***	8D3B00
Associated class	IBMz_ComputerSystem		
NumberOfDedicatedCPs			
Description: Number of dedicated standard processors.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors dedicated (CP)	***DDS metric ID***	8D3B20
Associated class	IBMz_ComputerSystem		
NumberOfDedicatedIIPs			
Description: Number of dedicated processors or cores of type zIIP.			

Property type	Property value	Property type	Property value
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors dedicated (IIP)	***DDS metric ID***	8D3B40
Associated class	IBMz_ComputerSystem		
NumberOfOnlineAAPs			
Description: Number of online processors or cores of type zAAP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (AAP)	***DDS metric ID***	8D3C70
Associated class	IBMz_ComputerSystem		
NumberOfOnlineCPs			
Description: Number of online standard processors.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (CP)	***DDS metric ID***	8D2610
Associated class	IBMz_ComputerSystem		
NumberOfOnlineICFs			
Description: Number of online processors of type ICF.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (ICF)	***DDS metric ID***	8D3CC0
Associated class	IBMz_ComputerSystem		
NumberOfOnlineIFLs			
Description: Number of online processors of type IFL.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)

Property type	Property value	Property type	Property value
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (IFL)	***DDS metric ID***	8D3D10
Associated class	IBMz_ComputerSystem		
NumberOfOnlineIIPs			
Description: Number of online processors or cores of type zIIP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# processors online (IIP)	***DDS metric ID***	8D3D40
Associated class	IBMz_ComputerSystem		
NumberOfSharedAAPs			
Description: Number of shared processors or cores of type zAAP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# logical processors shared (AAP)	***DDS metric ID***	8D3B60
Associated class	IBMz_ComputerSystem		
NumberOfSharedCPs			
Description: Number of shared standard processors (shared logicals).			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# logical processors shared (CP)	***DDS metric ID***	8D3B90
Associated class	IBMz_ComputerSystem		
NumberOfSharedIIPs			
Description: Number of shared processors or cores of type zIIP.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	# logical processors shared (IIP)	***DDS metric ID***	8D3C40
Associated class	IBMz_ComputerSystem		
PartitionCapacityCappedPercentage			
Description: WLM capping percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% WLM capping	***DDS metric ID***	8D2490
Associated class	IBMz_ComputerSystem		
PartitionCapacityFourHourAverage			
Description: Four hour rolling average.			
Datatype	uint64	***Units***	MSU/h
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	four hour MSU average	***DDS metric ID***	8D2630
Associated class	IBMz_ComputerSystem		
PartitionDefinedCapacity			
Description: Percentage of defined CPU capacity used by this LPAR or VM guest.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	defined MSU	***DDS metric ID***	8D2620
Associated class	IBMzOS_ComputerSystem		
PartitionDefinedCapacityUsedPercentage			
Description: Percentage of defined CPU capacity used by this LPAR or z/VM guest.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% capacity used	***DDS metric ID***	8D2870

Property type	Property value	Property type	Property value
Associated class	IBMzOS_ComputerSystem		
TotalAAPTimePercentage			
Description: zAAP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% total logical utilization (AAP)	***DDS metric ID***	8D38D0
Associated class	IBMz_ComputerSystem		
TotalAAPonCPTimePercentage			
Description: zAAP on CP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% AAP on CP	***DDS metric ID***	8D2C90
Associated class	IBMz_ComputerSystem		
TotalCPTimePercentage			
Description: Total CP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% total logical utilization (CP)	***DDS metric ID***	8D2510
Associated class	IBMz_ComputerSystem		
TotalIIPonCPTimePercentage			
Description: zIIP on CP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% IIP on CP	***DDS metric ID***	8D3550
Associated class	IBMz_ComputerSystem		

Property type	Property value	Property type	Property value
TotalIIPTimePercentage			
Description: Total zIIP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% total logical utilization (IIP)	***DDS metric ID***	8D3900
Associated class	IBMz_ComputerSystem		
RemainingTimeUntilCapping			
Description: Remaining time until capping in seconds.			
Datatype	unit64	***Units***	seconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	remaining time until capping in seconds	***DDS metric ID***	8D2680
Associated class	IBMz_ComputerSystem		
RemainingTimeUntilGroupCapping			
Description: Remaining time until group capping in seconds.			
Datatype	unit64	***Units***	seconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	remaining time until group capping in seconds	***DDS metric ID***	8D4460
Associated class	IBMz_ComputerSystem		
AvailableCapacityForCapacityGroup			
Description: available capacity (MSU/h) for the group.			
Datatype	unit64	***Units***	MSU/h
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	available capacity (MSU/h) for group	***DDS metric ID***	8D43E0
Associated class	IBMz_ComputerSystem		
ActualMSU			
Description: actual MSU.			

Property type	Property value	Property type	Property value
Datatype	unit64	***Units***	MSU/h
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	actual MSU	***DDS metric ID***	8D25E0
Associated class	IBMz_ComputerSystem		
DefinedCapacityGroupLimit			
Description: Defined capacity group limit (MSU/h).			
Datatype	unit64	***Units***	MSU/h
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	defined capacity group limit	***DDS metric ID***	8D4530
Associated class	IBMz_ComputerSystem		

IBMzOS_OperatingSystem

This class contains basic properties of a running z/OS operating system image.

Property type	Property value	Property type	Property value
DelayForAAPPercentage (breakdown dimension: WLM service class period)			
Description: Percentage of samples where the reported WLM service class period has been found delayed for a zAAP.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of zAAP delay samples by WLM service class period	***DDS metric ID***	8D37E0
Associated class	IBMzOS_OperatingSystem		
DelayForCPPercentage (breakdown dimension: WLM service class period)			
Description: Percentage of samples where the reported WLM service class period has been found delayed for a standard processor.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of standard CP delay samples by WLM service class period	***DDS metric ID***	8D3740
Associated class	IBMzOS_OperatingSystem		
DelayForIIPPercentage (breakdown dimension: WLM service class period)			
Description: Percentage of samples where the reported WLM service class period has been found delayed for a zIIP.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of zIIP delay samples by WLM service class period	***DDS metric ID***	8D3830
Associated class	IBMzOS_OperatingSystem		
DelayForProcessorPercentage (breakdown dimension: WLM service class period)			
Description: Total percentage of samples where the reported WLM service class period has been found delayed for any used processor type.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	% of total delay samples by WLM service class period	***DDS metric ID***	8D3790
Associated class	IBMzOS_OperatingSystem		
DelayPercentage (breakdown dimension: WLM service class period)			
Description: Percentage of total delay samples.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% delay by WLM service class period	***DDS metric ID***	8D17E0
Associated class	IBMzOS_OperatingSystem		
DelayCount (breakdown dimension: WLM service class period)			
Description: Number of samples where the reported WLM service class period has been found delayed.			
Datatype	real32	***Units***	samples
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# of total delay samples by WLM service class period	***DDS metric ID***	8D43C0
Associated class	IBMzOS_OperatingSystem		
FreePhysicalMemory			
Description: Number of KBytes of physical memory currently unused and available. On z/OS, physical memory is also known as central storage.			
Datatype	uint64	***Units***	kilobytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# frames available	***DDS metric ID***	8D2EE0
Associated class	IBMzOS_OperatingSystem		
FreeSpaceInPagingFiles			
Description: Total number of free Kbytes in the operating system's paging files. For z/OS, this is the number of slots available in z/OS auxiliary storage.			
Datatype	uint64	***Units***	kilobytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	# slots available	***DDS metric ID***	8D2F10
Associated class	IBMzOS_OperatingSystem		
FreeVirtualMemory			
Description: Number of KBytes of virtual memory currently unused and available.			
Datatype	uint64	***Units***	kilobytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# frames and slots available	***DDS metric ID***	8D2ED0
Associated class	IBMzOS_OperatingSystem		
InternalViewAAPTimePercentage			
Description: MVS view of the zAAP utilization.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% MVS utilization (zAAP)	***DDS metric ID***	8D3AE0
Associated class	IBMzOS_OperatingSystem		
InternalViewIIPTimePercentage			
Description: MVS view of the zIIP utilization.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% MVS utilization (zIIP)	***DDS metric ID***	8D3AF0
Associated class	IBMzOS_OperatingSystem		
InternalViewTotalCPUTimePercentage			
Description: MVS view of the CP utilization, only available for LPARs in which z/OS RMF is active.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% MVS utilization (CP)	***DDS metric ID***	8D0420

Property type	Property value	Property type	Property value
Associated class	IBMzOS_OperatingSystem		
KernelModeTime			
Description: Time in kernel mode on the operating system level. On z/OS, this is mapped to uncaptured time.			
Datatype	uint64	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	uncaptured time	***DDS metric ID***	8D3240
Associated class	IBMzOS_OperatingSystem		
LoadAverage			
Description: Average in-ready queue length, that is, the number of workloads queued to be executed on a CP processor. In relation to the number of available CPs, this metric can be used as a CPU contention indicator, because workloads waiting in the in-ready queue are delayed for CPU resources.			
Datatype	real32	***Units***	queue length
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	load average	***DDS metric ID***	8D30E0
Associated class	IBMzOS_OperatingSystem		
LocalPI (breakdown dimension: WLM service class period)			
Description: Local performance index; the performance index of the MVS image on which the contacted CIM provider is running.			
Datatype	real32	***Units***	n/a
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	performance index by WLM service class period	***DDS metric ID***	8D1020
Associated class	IBMzOS_OperatingSystem		
NumberOfProcesses			
Description: Number of z/OS address spaces active during the sample interval.			
Datatype	uint32	***Units***	processes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# users	***DDS metric ID***	8D0D50

Property type	Property value	Property type	Property value
Associated class	IBMzOS_OperatingSystem		
NumberOfUsers			
Description: Number of user sessions for which the operating system is currently storing state information. On z/OS, this is mapped to the number of TSO users currently active on the z/OS image.			
Datatype	uint32	***Units***	users
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	# users (Work scope: WLM Workload: TSO)	***DDS metric ID***	8D0D50
Associated class	IBMzOS_OperatingSystem		
OperationalStatus			
Description: Overall status of the associated CIM_OperatingSystem resource (= z/OS image). Currently, two states can be shown by the z/OS implementation: • 'Stressed' (= 0x10 as metric value), indicating that the system is severely overloaded • 'Ok' (= 0x04 as metric value). If the workflow with workscope SYSTEM is less than 60%, the system is reported to be 'Stressed', because in this case, even the workloads with highest priority are significantly delayed.			
Datatype	uint32	***Units***	n/a
ChangeType	n/a	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	false
DDS metric name	% workflow (based on DDS metric 0x8D0550 with work scope: WLM Workload: SYSTEM)	***DDS metric ID***	8D0550
Associated class	IBMzOS_OperatingSystem		
PageInRate			
Description: Number of pages paged in per second.			
Datatype	uint64	***Units***	events per second
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	pagein rate	***DDS metric ID***	8D30F0
Associated class	IBMzOS_OperatingSystem		
PercentDelay (z/OS only)			
Description: Percentage of samples during which the z/OS operating system was delayed for some resources.			
Datatype	real32	***Units***	percent

Property type	Property value	Property type	Property value
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% delay	***DDS metric ID***	8D0160
Associated class	IBMzOS_OperatingSystem		
RGCappingDelaySamples (breakdown dimension: WLM service class period)			
Description: Resource group capping delay samples percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% of RG capping delay samples by WLM service class period	***DDS metric ID***	8D3880
Associated class	IBMzOS_OperatingSystem		
SRBTimePercentage (breakdown dimension: WLM service class period)			
Description: Percentage of SRB time used by all work in the system, or by WLM classes. This metric is divided by the number of processors. It does not include zAAP and zIIP times.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% SRB by WLM service class period	***DDS metric ID***	8D2D40
Associated class	IBMzOS_OperatingSystem		
SysplexPI (breakdown dimension: WLM service class period)			
Description: Sysplex performance index.			
Datatype	real32	***Units***	n/a
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	performance index by WLM service class period	***DDS metric ID***	8D1020
Associated class	IBMzOS_OperatingSystem		
TCBTimePercentage (breakdown dimension: WLM service class period)			
Description: Percentage of TCB time used by all work in the system, or by WLM classes. This metric is divided by the number of processors. It does not include zAAP and zIIP times.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)

Property type	Property value	Property type	Property value
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% TCB by WLM service class period	***DDS metric ID***	8D2D50
Associated class	IBMzOS_OperatingSystem		
TotalAAPTimePercentage			
Description: zAAP utilization percentage			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% CPU utilization (zAAP)	***DDS metric ID***	8D39B0
Associated class	IBMzOS_OperatingSystem		
TotalAAPTimePercentage (breakdown dimension: WLM service class period)			
Description: zAAP utilization percentage for a given WLM service class period)			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% zAAP	***DDS metric ID***	8D2C60
Associated class	IBMzOS_OperatingSystem		
TotalAAPonCPTimePercentage (breakdown dimension: WLM service class period)			
Description: Total zAAP on CP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% zAAP on CP by WLM service class period	***DDS metric ID***	8D2D00
Associated class	IBMzOS_OperatingSystem		
TotalCPUTime			
Description: Total system CPU time used.			
Datatype	uint64	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	total time	***DDS metric ID***	8D31F0
Associated class	IBMzOS_OperatingSystem		
TotalIIPonCPTimePercentage (breakdown dimension: WLM service class period)			
Description: zIIP on CP time percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% zIIP on CP by WLM service class period	***DDS metric ID***	8D35D0
Associated class	IBMz_OperatingSystem		
TotalIIPTimePercentage			
Description: zIIP utilization percentage.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% CPU utilization (zIIP)	***DDS metric ID***	8D39C0
Associated class	IBMzOS_OperatingSystem		
TotalIIPTimePercentage (breakdown dimension: WLM service class period)			
Description: zIIP utilization percentage for a given WLM service class period)			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% CP utilization (zIIP)	***DDS metric ID***	8D3520
Associated class	IBMzOS_OperatingSystem		
UserModeTime			
Description: Time in user mode on operating system level. On z/OS, this is mapped to captured time.			
Datatype	uint64	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	captured time	***DDS metric ID***	8D3030
Associated class	IBMzOS_OperatingSystem		

IBMzOS_ComputerSystem

This class provides basic computer system information such as computer name, and status information. It represents either virtual or physical computer systems in the sense of a container inside which an operating system runs. On zSeries, this class represents either an LPAR or a z/VM® guest.

Property type	Property value	Property type	Property value
ActiveVirtualProcessors			
Description: Average number of regular CPs assigned to this IBMzOS_ComputerSystem (LPAR). If z/OS is running in a z/VM guest environment, this metric returns zero.			
Datatype	real32	***Units***	processors
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	average number of logical processors	***DDS metric ID***	8D2610
Associated class	IBMzOS_ComputerSystem		
PartitionDefinedCapacityUsedPercentage			
Description: Percentage of defined capacity actually used by the partition. If z/OS is running in a z/VM guest environment, this metric returns zero.			
Datatype	uint32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% capacity used	***DDS metric ID***	8D2870
Associated class	IBMzOS_ComputerSystem		

IBMzOS_Process

This class provides basic process information such as process name, priority, and run-time state. Instances of class IBMzOS_Process are mapped to z/OS address spaces. Client applications can use this class to give clients an understanding of the processes (address spaces) running on the managed system within the context of their operating system.

Property type	Property value	Property type	Property value
PageInRate			
Description: Number of pages paged in per second on behalf of the associated process.			
Datatype	uint64	***Units***	events per second
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	pagin rate per residency time by job	***DDS metric ID***	8D1090
Associated class	IBMzOS_Process		
ResidentSetSize			
Description: Memory in bytes currently allocated in physical memory by the given process. Also known as 'working set size'.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	working set by job	***DDS metric ID***	8D1280
Associated class	IBMzOS_Process		
TotalCPUTime			
Description: Amount of CPU time used by the given process.			
Datatype	uint64	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	eappl time by job	***DDS metric ID***	8D3070
Associated class	IBMzOS_Process		
UserModeTime			
Description: CPU time used in user mode specifically for the given process. On z/OS this is currently the same as **TotalCPUTime**.			
Datatype	uint64	***Units***	milliseconds
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true

Property type	Property value	Property type	Property value
DDS metric name	eappl time by job	***DDS metric ID***	8D3070
Associated class	IBMzOS_Process		

IBMzOS_UnixProcess

This class provides basic information about z/OS processes running in the z/OS UNIX System Services. It implements all properties from CIM_Process plus a set of properties typical for UNIX processes.

Property type	Property value	Property type	Property value
AccumulatedTotalCPUTime			
Description: CPU time in seconds spent for this process since USS process creation.			
Datatype	real32	***Units***	seconds
ChangeType	3 (Counter)	***TimeScope***	4 (Startup Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	total cpu seconds by uss pid and jobname	***DDS metric ID***	8D31E0
Associated class	IBMzOS_UnixProcess		
ExternalViewUserModePercentage			
Description: Usage percentage of the system CPUs for this process in user mode during the measurement interval.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% appl by uss pid and jobname	***DDS metric ID***	8D2830
Associated class	IBMzOS_UnixProcess		

IBMzOS_Channel

Instances of this class represent z/OS channels, with property DeviceID containing the Channel Path ID (CHPID). The purpose of this resource class is to enable the association of related metrics.

Property type	Property value	Property type	Property value
BusUtilization			
Description: Percentage of bus cycles, the bus has been found busy for this channel in relation to the theoretical limit.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% bus utilization	***DDS metric ID***	8D2360
Associated class	IBMzOS_Channel		
BytesReceived			
Description: Total number of bytes received per second, including framing characters			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	total bytes read/sec	***DDS metric ID***	8D23D0
Associated class	IBMzOS_Channel		
BytesTransmitted			
Description: Total number of bytes transmitted per second, including framing characters.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	total bytes written/sec	***DDS metric ID***	8D23F0
Associated class	IBMzOS_Channel		
ErrorRate			
Description: Number of network errors per second.			
Datatype	real32	***Units***	errors per second
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	send fail/sec	***DDS metric ID***	8D31C0

Property type	Property value	Property type	Property value
Associated class	IBMzOS_Channel		
NetworkPortUtilizationPercentage			
Description: Percentage of z/OS channel capacity actually used. This number may not be precise, meaning that for some non-ideal work loads (like transferring lots of very short messages), it may not be possible to reach 100% utilization.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% total utilization	***DDS metric ID***	8D0080
Associated class	IBMzOS_Channel		
PartitionBytesReceived (z/OS only)			
Description: Total number of bytes received.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	partition bytes read/sec	***DDS metric ID***	8D2390
Associated class	IBMzOS_Channel		
PartitionBytesTransmitted (z/OS only)			
Description: Total number of bytes transmitted.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	partition bytes written/sec	***DDS metric ID***	8D23B0
Associated class	IBMzOS_Channel		
PartitionUtilization			
Description: Channel path utilization percentage for an individual logical partition.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% partition utilization	***DDS metric ID***	8D0060

Property type	Property value	Property type	Property value
Associated class	IBMzOS_Channel		
ReceiveErrorRate (z/OS only)			
Description: Number of network errors per second related to receiving activities of the channel.			
Datatype	real32	***Units***	errors per second
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	receive fail/sec	***DDS metric ID***	8D3160
Associated class	IBMzOS_Channel		
TotalUtilization			
Description: Channel path utilization percentage for the entire system during the interval.			
Datatype	real32	***Units***	percent
ChangeType	4 (Gauge)	***TimeScope***	3 (Interval)
GatheringType	3 (Periodic)	***IsContinuous***	true
DDS metric name	% total utilization	***DDS metric ID***	8D0080
Associated class	IBMzOS_Channel		

IBMzOS_UnixLocalFileSystem

This class represents UNIX file systems that are locally attached to a computer system. On z/OS, the hierarchical file system zFS is supported.

Note: The class IBMzOS_UnixLocalFileSystem is inherited from CIM_FileSystem.

Property type	**Property value**	**Property type**	**Property value**
AvailableSpace			
Description: Total amount of free space for the associated file system in bytes. This metric is not implemented based on RMF data.			
Datatype	uint64	***Units***	bytes
ChangeType	4 (Gauge)	***TimeScope***	2 (Point)
GatheringType	4 (OnRequest)	***IsContinuous***	true
DDS metric name	none	***DDS metric ID***	none
Associated class	IBMzOS_UnixLocalFileSystem		

Chapter 3. Adding Monitor I and Monitor II installation exits

This chapter describes:

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

Overview

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

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

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

During a Monitor II TSO/E display session, with TSO/E installed, a user exit enables your installation to verify that a terminal user is authorized to use RMF. See “TSO terminal user authorization” on page 105 for an explanation of this user exit.

Monitor I session user reports

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

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

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

Guidelines

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

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

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

Caution

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

Initialization for Monitor I session user exit routines

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

When the exit routine gets control, register 1 points to a three-word address list. The first address points to the two-word area reserved for use by your routines. This same two-word area is passed to all the user exit routines and can be used for communication between them. The second address points to the RMF phase parameter, a full-word field that is always X‘4:’, indicating that the exit is called during Monitor I session initialization. The third address points to a word that is relevant only when you are providing a routine to sample data at each cycle; one of the functions your initialization routine will perform is to put the address of the user sampler in this word. Figure 8 on page 87 illustrates the input parameter structure.

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

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

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

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

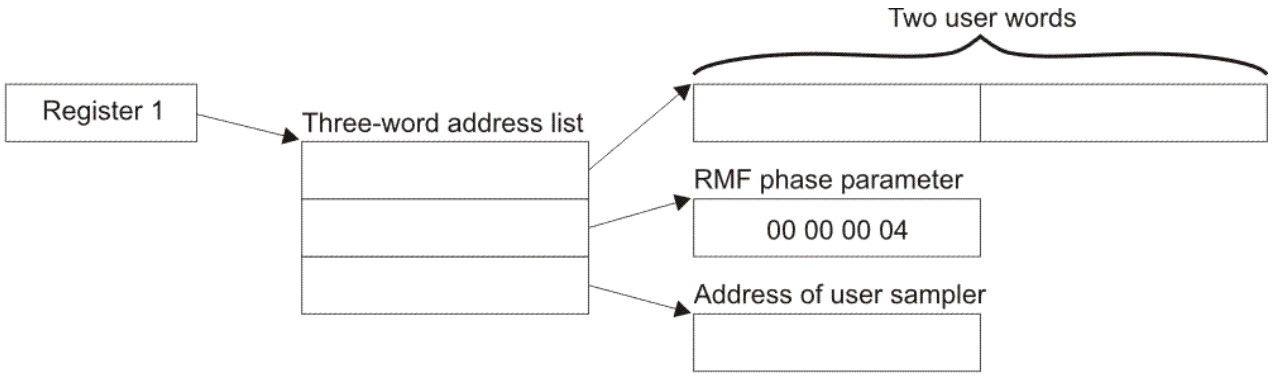


Figure 8. ERBMFIUC Input Parameter Structure

Sampling data at each cycle

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

When the sampler gets control, register 1 points to a two-word area. One of these words, selected by your installation, contains the address of the storage area obtained for the sampler by ERBMFIUC. Figure 9 on page 87 illustrates the input parameter structure.

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

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

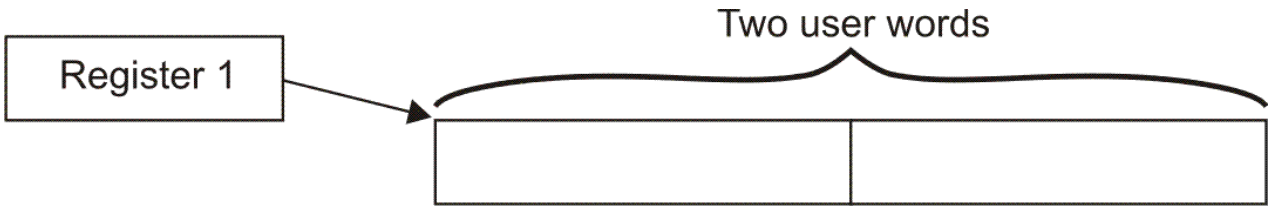


Figure 9. User Sampler Input Parameter Structure

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

Interval processing

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

When the exit gets control, register 1 points to a two-word address list. The first address points to the two-word area reserved for use by your routines. When these routines include a user sampler, one of these words, selected by your installation, will contain the address of the sampled data. The second address points to the RMF phase parameter. This parameter is a full word that contains X‘4’ when the exit is called during Monitor I session initialization, X‘8’ when the exit is called at the end of an RMF interval, or X‘C’ when the exit is called at the end of an RMF interval for which data collection was skipped. Figure 10 on page 88 illustrates the input parameter structure.

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

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

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

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

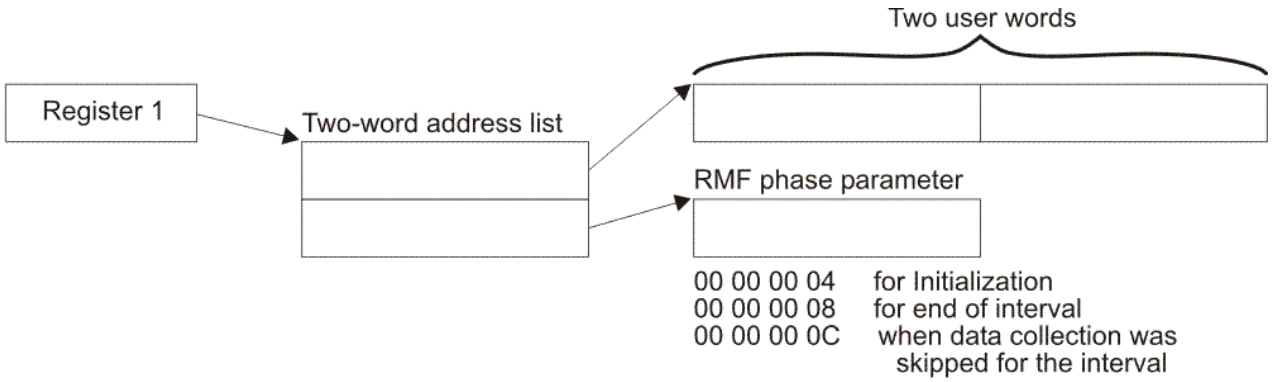


Figure 10. ERBMFDUC Input Parameter Structure

Report writing during session processing

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

When the exit gets control, register 1 points to a two-word address list. The first address points to the two-word area reserved for use by user routines. The second address points to the RMF phase parameter, which is always X'10' for the report writer. Figure 11 on page 89 illustrates the input parameter structure.

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

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

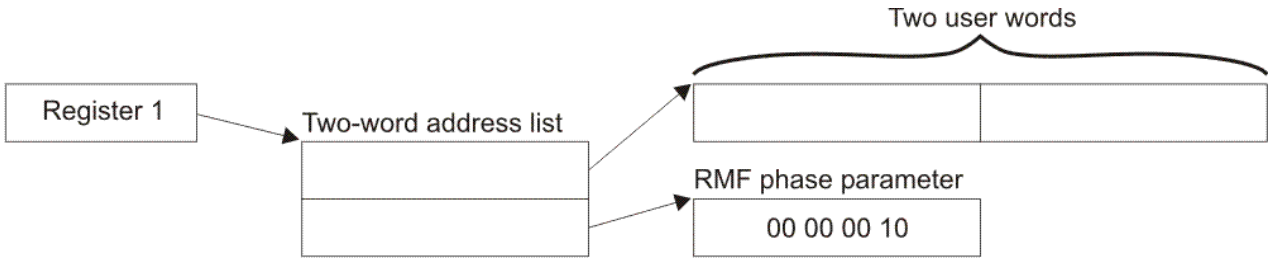


Figure 11. ERBMFRUR Input Parameter Structure

Termination

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

When the exit gets control, register 1 points to a two-word address list. The first address points to a two-word area reserved for use by your routines. The second address points to the RMF phase parameter, which is always X'C' for termination. Figure 12 on page 89 illustrates the input parameter structure.

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

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

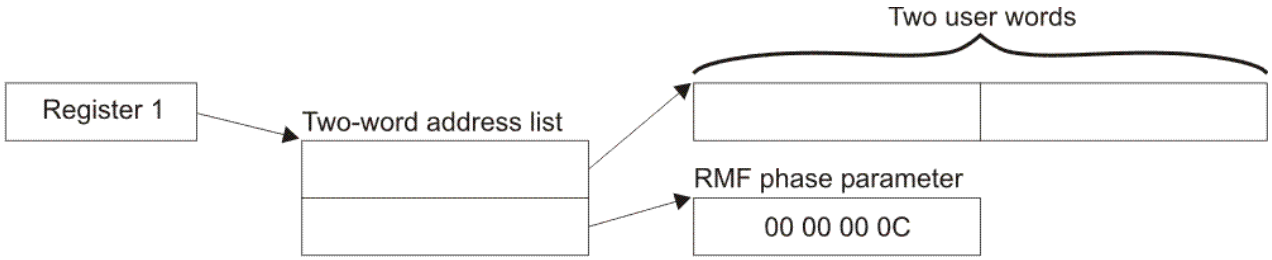


Figure 12. ERBMFTUR Input Parameter Structure

Tracing your own field

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

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

Step 1 -- Defining the name to RMF

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

1. The name of the field to be traced. The name can be from one to eight bytes long. It must not be the same as any name already recognized by RMF. When the name of the field is less than eight bytes long, it must be padded on the right with blanks to a length of eight bytes.
2. The length of the name. This field is one byte long; the value must be from 01 to 08.

3. A one-byte constant that always contains the value X'DC'.
4. The length of the field to be traced. This field is one byte long; the value must be from 01 to 04.

One byte of binary zeroes must follow the last entry to be added; the byte of binary zeroes indicates the end of the variable-length trace table. Figure 13 on page 90 shows an example of how to superzap ERBMFTTB to add a new name for tracing. The parenthesized numbers in the text refer to the parenthesized numbers in the figure. The example adds a nucleus field named MYDATA (1) that is two bytes in length (4) to the list of names valid for RMF tracing. The name is six bytes long (2), and the required constant is also supplied (3). A byte of binary zeroes (5) indicates the end of the trace table. Adding the name definition to ERBMFTTB causes RMF to pass the name to ERBTRACE during each tracing cycle. The four fields must be added for each name you want RMF to trace; only the last entry must be followed by the byte of binary zeroes.

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

Figure 13. Example of Adding a Name to ERBMFTTB

Step 2 -- Replacing ERBTRACE

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

When ERBTRACE gets control, register 1 points to a two-word address list. The first address points to an eight-byte field that holds the name to be validated. The second address points to a full word to be used by ERBTRACE to return the address of the user field to RMF. Figure 14 on page 90 illustrates the input parameter structure.

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

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

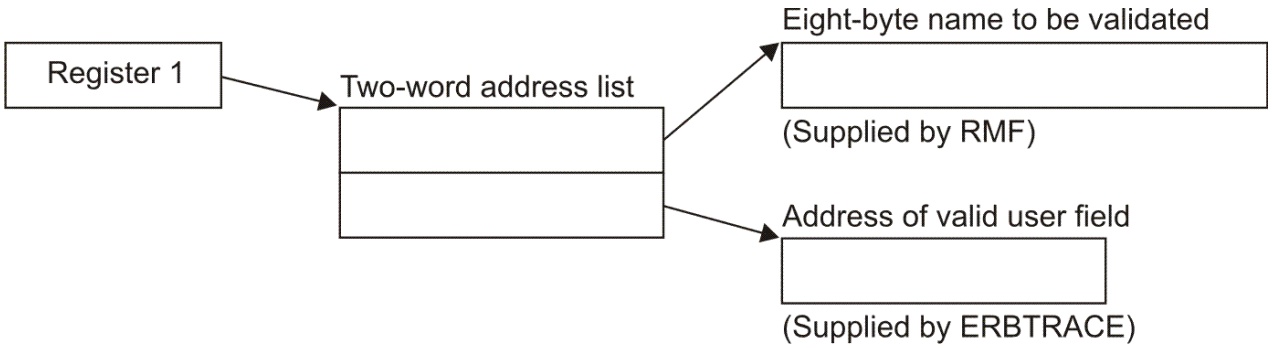


Figure 14. ERBTRACE Input Parameter Structure

Report writing by the Postprocessor

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

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

Figure 15 on page 91 illustrates the input parameter structure.

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

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

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

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

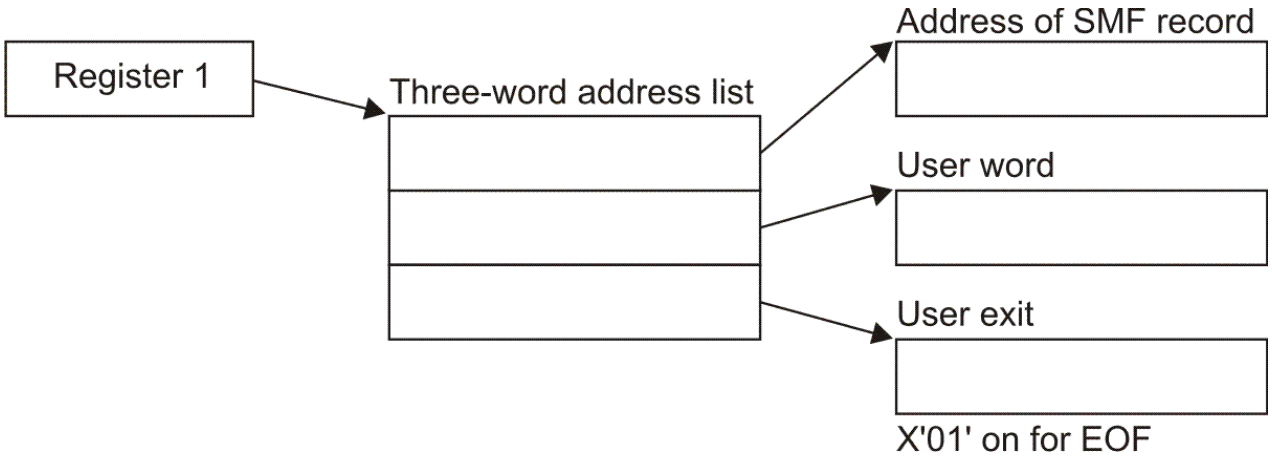


Figure 15. ERBMFPUS Input Parameter Structure

Adding your routines to RMF

Before your Monitor I session user exit routines can be tested and used, they must be assembled and link edited with the appropriate RMF modules. If you are using your private libraries, you have to ensure that they are concatenated in front of the distributed RMF libraries. Figure 16 on page 92 shows sample JCL for performing the required link edit for all user routines except the sampler routine. If you have a user sampler, a separate link edit is required; a sample is shown in Figure 17 on page 92.

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

Figure 16. Replacing Installation Exits

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

Figure 17. Adding a User Sampler

Monitor II session user reports

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

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

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

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

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

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

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

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

Guidelines

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

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

SMF record type 79

SMF record type 79 must be used to record data gathered by a user data gathering routine. Figure 18 on page 94 shows the layout of the record sections that are common to all Monitor II data gatherers, whether coded by a user or provided by RMF. The figure illustrates the layout of these common sections by showing the expansion of the RMF mapping macro ERBSMF79.

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

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

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

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

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

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

Figure 18. ERBSMF79 Mapping Macro Expansion

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

Figure 19. ERBSMF79 Mapping Macro Expansion (continued)

Relocate blocks

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

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

When you add a menu item to ERBFMENU or an option to ERBBMENU, the entry that describes the new report must include a field that specifies the length of the relocate block, the maximum number of possible relocate blocks, and the length of the data control section. For information on how to add an entry to ERBFMENU or ERBBMENU, see “Using the PICTURE macro” on page 104. To determine the storage to allocate, RMF multiplies the length of the relocate block by the maximum number of relocate blocks and adds this value to the length of the data control section and the common section. The result of this computation is the maximum possible length of the SMF record, and RMF allocates a buffer for the record that is equal in size to the maximum length.

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

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

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

Data control section

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

1. Set the value for the FBLEN parameters on the PICTURE macro instruction for your report, as described under “Using the PICTURE macro” on page 104.
2. Format the data control section to hold the common data.
3. Place it between the Monitor II control section and the data section. SMF79DCS contains the offset at which it should start.
4. Set SMF79DCL and SMF79DCN to the length and number of the data control sections.
5. Set the offset to the first data section SMF79ASS to point to the end of the data control section.

When a data control section is *not* used:

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

Coding a user report

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

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

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

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

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

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

Data gatherer

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

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

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

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

Operand 1	Operand 2
LL	text

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

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

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

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

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

Operand 1	Operand 2
LL	text

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

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

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

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

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

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

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

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

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

Your routine should complete the required fields in the SMF record common section (the B fields in Figure 18 on page 94), using the RMF mapping macro ERBSMF79 to access the fields in the common section. The address of the storage buffer obtained for your record is passed in the fourth input parameter. Your routine would gather the data required and format the data section of the record as agreed upon by convention between the data gatherer and the data reporter. Should your routine locate no data that is applicable to the report requested, set a return code of 16 in register 15 and return control.

When your routine has finished processing, set a return code in register 15 and return to the caller by branching on the contents of register 14. Table 4 on page 98 shows the possible return codes, their meaning, and the action RMF takes in response. These return codes apply to both the data gatherer and the data reporter.

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

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

Code	Meaning	RMF Response (Display Session)	RMF Response (Background Session)
0	Successful completion.	The session continues.	The session continues.
4	Invalid operand syntax.	The command is displayed as entered.	Message ERB409I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues. The operator can modify the session options.
8	Invalid entry code.	Abend - the user code is 1402.	Abend - the user code is 1402.
12	I/O error.	Messages ERB403I and ERB404I are displayed, including the SYNAD text.	The current measurement continues when RECORD is in effect, but no subsequent reports are printed; otherwise, the measurement is discontinued. The session continues.
16	No data found.	Message ERB405I is displayed.	Message ERB405I is issued. No report or SMF record is produced for this interval. All measurements continue.
20	ESTAE macro failed.	Message ERB406I is displayed.	Message ERB406I is issued. The current measurement continues if the error was detected by the data reporter and RECORD is in effect; otherwise, the measurement is discontinued. The session continues.

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

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

Data reporter

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

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

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

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

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

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

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

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

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

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

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

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

Operand 1	Operand 2
LL	text

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

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

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

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

Operand 1	Operand 2
LL	text

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   'HD'
   Indicates that the record is a header line

   'DT'
   Indicates that the record is a data line

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

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

   Bit
   Meaning

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

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

   Fourth Putline Parameter: The control block address that RMF passed to your data reporter in the eleventh input parameter.

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

When your data reporter has finished processing, set a return code in register 15 and return control by branching on the contents of register 14. Table 4 on page 98 shows the possible return codes, their meaning, and the action RMF takes in response to each code.

Installing a user report

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

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

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

   RMF supplies the PICTURE macro to simplify the process of adding or changing an entry in the option list or menu list. See “Using the PICTURE macro” on page 104. You can also superzap an entry to make changes when the length of the entry is not changed.
2. Link edit your data gatherer and data reporter and test your report.

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

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

1. Determine whether the USER entry supplied by RMF is appropriate for your report. The USER entry contains specifications for a table report (RPTTYP=T) with a single relocate block (MAXRBS=1) that is four bytes long (RBLEN=4). The report title is ‘USER PICTURE’. If the entry is not appropriate for your report, replace the entry with a new entry for USER.
2. If you are changing the USER entry or adding a new entry, make a copy of ERBFMENU for a display report or ERBBMENU for a background session — or both — from the source code data set.
3. In the copy you have made, either replace the USER entry or insert a new PICTURE macro. For a new display report, insert the PICTURE macro where you want the new report to appear in the menu frame. For details, see “Using the PICTURE macro” on page 104.
4. Assemble ERBFMENU for a display report and ERBBMENU for a background report.
5. Link edit the menu list or option list CSECT(s) that you have assembled into the RMF load modules:

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

   A sample of the control statements required is:

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

Figure 20. Install User Report

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

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

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

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

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

1. Use a testing tasklib, a special partitioned data set (for example, TESTLIB.LOAD). Place your data gatherer, data reporter, and the RMFMON load module that includes the new menu list in the testing tasklib.
2. You can then test the new report by entering:

   ```
   CALL TESTLIB(RMFMON)
   ```

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

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

Using the PICTURE macro

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

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

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

Figure 21. Syntax of the PICTURE Macro

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

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

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

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

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

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

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

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

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

DRTEXT='drdefaults'
The default operands that are passed to the data-reporting routine for the report. This field is optional; it is used when the report requires operands. The text must be enclosed in single quotation

marks. Use two single quotation marks to represent any quotation marks used in the text. Any characters are valid between the quotation marks. When more than 32 characters are specified, the text is truncated.

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

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

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

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

Example

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

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

TSO terminal user authorization

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

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

ERBTSOCK (Monitor II) runs in problem state with a key of 8. When this control section gets control, register 1 points to a two-word address list. The first address points to the eight-byte userid of the user who has issued the RMFMON command. The second word points to the PSCB. Figure 22 on page 105 illustrates the input parameter structure.

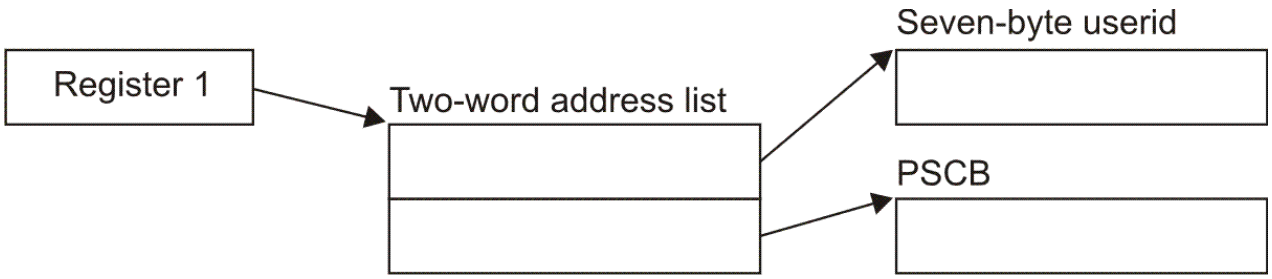


Figure 22. ERBTSOCK Input Parameter Structure

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

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

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

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

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

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

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

Chapter 4. Adding Monitor III user exits

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

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

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

Overview

The **RMF Monitor III Utility** (see "The Monitor III utility" on page 113) is the most important tool at your disposal for writing user exits. It is dialog-driven, and helps you use the necessary ISPF table services and RMF data-retrieval interface. However, you should be familiar with ISPF, Dialog Management Services, and RMF if you want to create and implement your own exit routines.

Data gathering

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

Reporting

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

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

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

the ISPF tables with the report data. The Monitor III Utility helps you to modify phases 1 and 3. Phases 2 and 4 are provided specially for user reports. See "Data reporter phases" on page 113 for more details.

Invoking user reports

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

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

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

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

"Data retrieval service (ERB3RDRS)" on page 137 describes this process.

Measurement data

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

Your user exits can use this means of communication, too. The format of the sample data is described in “Data gatherer sample structure” on page 108.

Data gatherer sample structure

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

Figure 23 on page 109 shows the layout of three data areas that are common to all Monitor III data gatherers, whether coded by a user or provided by RMF. These areas are:

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

Field offsets in the sample header and resource data record refer to offsets from the start of the control block containing the field. For example, the address of the first user record is the address of the REDG3 plus the offset to the first user record. All of these areas are maintained by RMF, specifically by the mainline data gathering module (ERB3GMFC). Figure 23 on page 109 also shows the relationship between the data collected by the data gatherer user exit routine and the sample structure maintained by RMF.

Note: For a description of how Monitor III maintains a set of samples when VSAM data sets are used with data set support, see "Using Monitor III VSAM data set support" in *z/OS RMF Data Gatherer Programmer's Guide*.

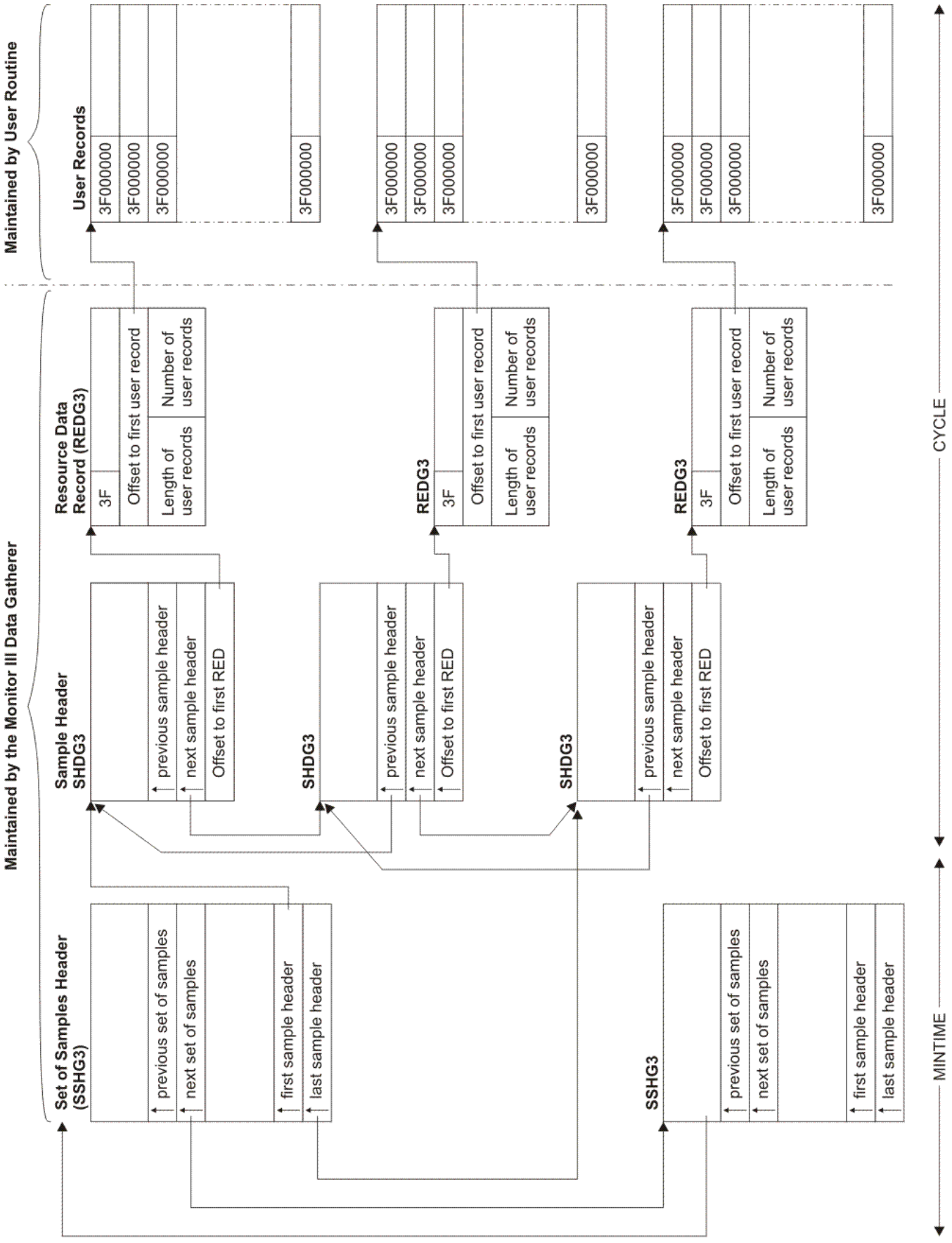


Figure 23. Data Gatherer Sample Structure

Data gatherer control blocks

Figure 24 on page 110 describes the fields in the set-of-samples header control block, the sample header, and the resource data record. These data areas are mapped by the RMF macros ERBSSHG3, ERBSHDG3, and ERBREDG3.

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

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

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

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

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

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

Figure 24. Mapping Macros of ERBSSHG3, ERBSHDG3 and ERBREDG3

Set of samples header control block (SSHG3)

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

Sample header control block (SHDG3)

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

Resource data record (REDG3)

There is one resource data (RED) record for each defined resource in the system. RMF maintains and updates all fields in this record as needed. RMF uses RED records to access USE/WAIT records (in the case of the Monitor III data gatherer) or user records (in the case of a data gathering user exit routine).

RED records are fixed in length, and contain X'3F' in the resource identifier (REDREDID) field when RMF invokes your data gatherer user exit routine. RMF uses this identifier to locate your user records, which also must have the same hexadecimal identifier. The RED record also contains the offset to the first user record (REDFUWDO), the length of your user exit records (REDUSERL), and the number of user exit records (REDUSERN) created during a CYCLE. While RMF maintains all the fields in the RED record, it obtains the length and number of user records from values you provide in the interface area used by the Monitor III data gatherer and your user routine. When RMF invokes your user exit, the second input parameter points to this interface area (see "Programming a data gatherer" on page 111).

User record

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

Programming a data gatherer

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

Register 13
Address of the register save area

Register 14
Return address

Register 15
Entry address

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

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

GGDMODNA
The module name, which is ERB3GUSR.

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

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

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

GGDAULEN
The length of the storage area.

GGDAUPTR
The address of the storage area.

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

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

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

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

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

  RETRUBFL
  Specifies registers that must be restored by the recovery termination manager (RTM) before returning control to the address in your routine specified in the RETADDR field. This field should contain X‘FFFF’, indicating that all registers must be restored after must be restored after error-recovery processing completes.

  RETREGSA
  A 16-word storage area used to store the contents of the registers specified in the RETRUBFL field.
- Third fullword – The address of an area containing the data the user exit routine collects. Your routine must supply this address each time it is invoked. RMF uses this address to move the collected data from the exit routine's storage area into the data gatherer's in-storage buffer.
- Fourth fullword – Two halfwords that the user exit routine must provide at each invocation. The first halfword must contain the length of the user record, and the second halfword must contain the number of user records collected during the current cycle. RMF places the length and number of user records in the resource data (RED) record. All user records must be fixed in length and must start with a fullword hexadecimal identifier of X‘3F000000’. RMF uses this information to move your collected data into the in-storage buffer.

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

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

The processing your data gathering routine performs depends largely on the nature of the report for which you are gathering data. The first time RMF invokes your data gatherer routine, it provides a subpool number (in the GGDAUSBP field) that you must use when issuing the GETMAIN macro. After issuing a

GETMAIN for the dynamic storage it needs to execute in, your routine must place the address and length of the storage obtained in the GGDAUPTR and GGDAULEN fields, respectively. (The GGDAUPTR and GGDAULEN fields contain zeroes when RMF invokes your routine for the first time.) When RMF makes subsequent calls to your routine, these two fields still contain the address and length of your dynamic storage. You do not have to issue another GETMAIN and you can reuse the storage obtained on the first call. This function eliminates the overhead of issuing a GETMAIN for dynamic storage each time RMF invokes your routine. Depending on the amount of data you collect, you may need to obtain additional storage to hold your user records.

Data reporter phases

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

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

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

- **Phase 1:** RMF generates an ISPF table that contains display data for every modifiable RMF report. Chapter 5, “Monitor III data reporter tables,” on page 143 describes these tables. The time range for the display data for your routine can be changed during this phase by calling the Data Retrieval Service (ERB3RDRS) module. See “Data retrieval service (ERB3RDRS)” on page 137 for information about how to invoke the Data Retrieval Service.

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

Note: If you decide to replace any of these phases, you must conform to the standards and externals described in this manual. If you do not, the results are unpredictable. See “Installing your own phases” on page 134.

The Monitor III utility

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

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

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

Chapter 5, “Monitor III data reporter tables,” on page 143 contains samples of each table and its entries.

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

Report utility panel flow

Figure 25 on page 115 shows the panel sequence for the report format definition utility.

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

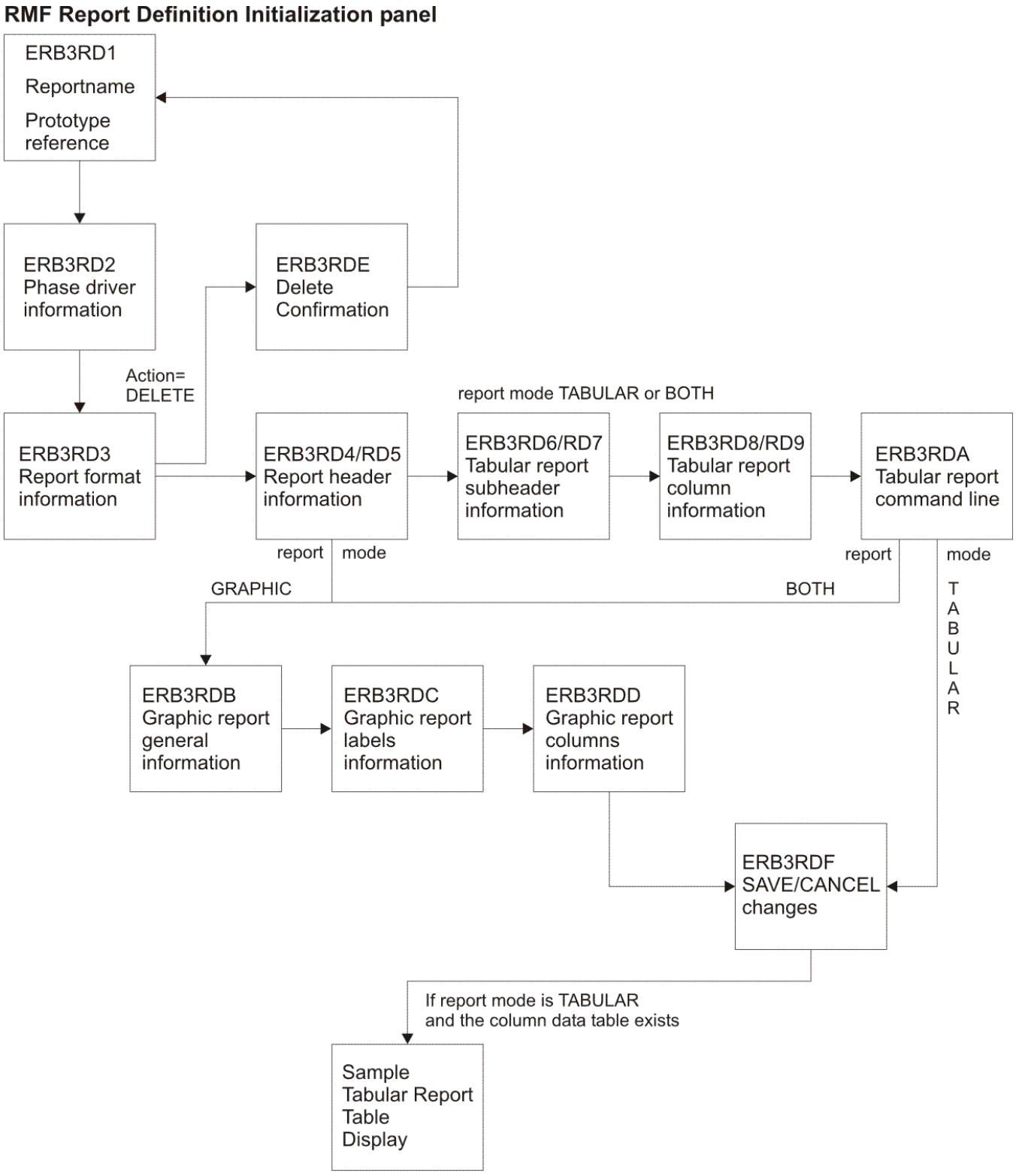


Figure 25. Panel Sequence for the Report Definition Utility

Before you start the utility

Note: The actions described in this section are only required if you do not want to use the standard concatenation of the RMF libraries.

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

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

You can merge your own libraries with RMF libraries. If you want to change the data set names and the allocations, modify CLIST ERBRMF3X. CLIST ERBRMF3X allocates the RMF ISPF libraries from the following distribution libraries:

- Panels from SYS1.SERBPENU
- Tables from SYS1.SERBTENU
- Messages from SYS1.SERBMENU

This CLIST is available in SYS1.SERBCLS, which must be concatenated to your SYSPROC library.

Starting the report utility

As a prerequisite for the invocation of the Monitor III report format definition utility, the SYS1.SERBCLS data set must be concatenated to your SYSPROC library. For more information, see *z/OS RMF Reporter User's Guide*.

To start the utility, use one of the following commands:

- from TSO/E ready mode: `RMF UTIL`
- from within ISPF: `TSO RMF UTIL`

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

```
RMFJPN UTIL
```

Note:

1. Do not use a 3270 session with a screen size lower than 32x80.
2. Do not try to access the report format definition utility in split screen mode when you are in an active RMF Monitor III reporter session.

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

Example - Modified SYSINFO report

The task of creating a new Monitor III report is shown based on the example of a modified SYSINFO report. The SYSINFO report has this format:

```
                        RMF V2R4  System Information            Line 1 of 28
Command ===>                                                  Scroll ===> HALF

Samples: 100     System: MVS3  Date: 05/11/19  Time: 10.03.20  Range: 100   Sec

Partition:    MVS1    2964 Model 757          Appl%:      63  Policy: STANDARD
CPs Online:    4.0    Avg CPU Util%:  73      EAppl%:     65  Date:   05/11/19
AAPs Online:     -    Avg MVS Util%:  84      Appl% AAP:   -  Time:   14.05.07
IIPs Online:   0.0                           Appl% IIP: 0.0

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

*SYSTEM      31  669   26        13.95  5.3  5.0   5.6  4.1  7.0  2.6  2.0  2.0
*TSO         50  534    8        13.95  2.6  2.1   0.4  1.5  2.0  0.8  0.0  0.0
*BATCH       26   11   10         0.00  1.5  1.4   1.4  1.7  0.5  1.8  1.0  2.0
*STC         27  115    8         0.00  1.1  1.5   0.1  1.0  4.5  0.1  1.0  0.0
*ASCH             3    0         0.00  0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0
*OMVS             2    0         0.00  0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0
*ENCLAVE      5    4  N/A          N/A  0.2  N/A   3.7  N/A  0.0  N/A  N/A  N/A
PRIMEBAT  W  26   11   10  46000  0.06  1.5  1.4   1.4  1.7  0.5  1.8  1.0  2.0
NRPRIME   S  26   11   10  46000  0.06  1.5  1.4   1.4  1.7  0.5  1.8  1.0  2.0
          1  23    9    9  27900  0.06  0.9  1.4   0.8  1.6  0.5  1.8  1.0  2.0
          2  29    0    0  54200  0.02  0.1  0.0   0.2  0.0  0.0  0.0  0.0  0.0
          3  59    1    1  0.000  0.00  0.6  0.0   0.4  0.0  0.0  0.0  0.0  0.0
PRIMETSO  W  50  527    8  759.0 13.98  2.6  2.1   0.4  1.5  2.0  0.8  0.0  0.0
TSOPRIME  S  50  527    8  759.0 13.98  2.6  2.1   0.4  1.5  2.0  0.8  0.0  0.0
          1  48  526    8  403.0 13.98  2.1  1.9   0.3  1.3  2.0  0.8  0.0  0.0
          2  75    1    1  30600  0.08  0.3  0.2   0.1  0.1  0.0  0.0  0.0  0.0
          3  75    0    0   126K  0.02  0.1  0.0   0.0  0.0  0.0  0.0  0.0  0.0
```

Figure 26. SYSINFO Report

The target is to create a report called SYSCPU that provides some more CPU related information as TCB% and SRB% for each group. This data is available in the corresponding Monitor III table.

```
                        RMF V2R4   System Information            Line 1 of 26
Command ===>                                                  Scroll ===> PAGE
Press END to return.
Samples:     60  System: SYSE  Date: 06/08/2019  Time: 08.59.00  Range: 60 Sec

Partition:    SYSE    2084  Model 314         Appl%:      22  Policy: STANDARD
CPs Online:   4.0     Avg CPU Util%:   24     EAppl%:     22  Date:   05/12/05
AAPs Online:  1.0     Avg MVS Util%:   26     Appl% AAP:  19  Time:   12.20.27
IIPs Online:    -                            Appl% IIP:   -

Group     T WFL --Users--   RESP TRANS  CPU   TCB   SRB  -AVG USG-  -Avg Del-
             %   TOT  ACT   Time  /SEC   %     %     %   PROC  DEV  PROC  DEV

*SYSTEM      98  167    2         0.12  21.6  21.5   0.2  1.6  0.1   0.0  0.0
*TSO              3    0         0.12   0.0   0.0   0.0  0.0  0.0   0.0  0.0
*BATCH       99    3    2         0.00  20.5  20.5   0.0  1.5  0.0   0.0  0.0
*STC         91  158    0         0.00   1.2   1.0   0.2  0.1  0.1   0.0  0.0
*ASCH             0    0         0.00   0.0   0.0   0.0  0.0  0.0   0.0  0.0
*OMVS             3    0         0.00   0.0   0.0   0.0  0.0  0.0   0.0  0.0
*ENCLAVE          0  N/A          N/A   N/A   N/A   N/A  0.0  N/A   0.0  N/A
BATCH     W  99    4    2  0.000  0.00  20.5  20.5   0.0  1.5  0.0   0.0  0.0
BATCHLOW  S  99    3    2  0.000  0.00  20.5  20.5   0.0  1.5  0.0   0.0  0.0
OMVSKERN  S        1    0  0.000  0.00   0.0   0.0   0.0  0.0  0.0   0.0  0.0
OMVS      W        2    0  0.000  0.00   0.0   0.0   0.0  0.0  0.0   0.0  0.0
OE        S        2    0  0.000  0.00   0.0   0.0   0.0  0.0  0.0   0.0  0.0
STC       W 100   26    0  0.000  0.00   0.1   0.1   0.0  0.0  0.0   0.0  0.0
STCDEF    S 100   26    0  0.000  0.00   0.1   0.1   0.0  0.0  0.0   0.0  0.0
SYSTEM    W  90  132    0  0.000  0.00   1.0   0.9   0.2  0.1  0.1   0.0  0.0
SYSSTC    S 100  111    0  0.000  0.00   0.3   0.3   0.0  0.0  0.0   0.0  0.0
SYSTEM    S  88   21    0  0.000  0.00   0.7   0.6   0.2  0.1  0.1   0.0  0.0
TSO       W        3    0  10.00  0.17   0.0   0.0   0.0  0.0  0.0   0.0  0.0
TSODEF    S        3    0  10.00  0.17   0.0   0.0   0.0  0.0  0.0   0.0  0.0
BCP       R  99    3    2  0.000  0.00  20.5  20.5   0.0  1.5  0.0   0.0  0.0
```

Figure 27. SYSCPU Report as Modification of the SYSINFO Report

You find details about all values that can be displayed for all Monitor III reports in Chapter 5, “Monitor III data reporter tables,” on page 143.

Report format definition panel (ERB3RD1)

After you call the report format definition utility by RMF UTIL, you get the **Report Definition Initialization** panel (ERB3RD1). On this panel, you can specify whether you want to create a new report or modify or delete an existing one. You can also select the name of an existing RMF report to use as a prototype for the new report.

```
ERB3RD1                 RMF Report Format Definition               Row 1 of 7
Command ===> _

Enter the following information.  To continue press ENTER.
To exit enter CANCEL or press the END key.

ACTION          ===> CREATE         MODIFY, CREATE or DELETE
REPORT NAME     ===> SYSCPU         Name of report
  WLM MODE      ===> GOAL           WLM Mode of report (GOAL or COMPAT)

Enter following information only, if you want to use an existing report
definition as a prototype for the new report you want to create.

PROTOTYPE NAME  ===> SYSINFO        Name of existing report to be used
  WLM MODE      ===> GOAL           WLM Mode of existing report to be used

 The following report names are available for MODIFY or as prototype

 CACHDET    CACHSUM    CFACT      CFOVER     CFSYS      CHANNEL
 CPC        CRYACC     CRYOVW     CRYPKC     DELAY      DEV
 DEVN       DEVR       DEVT       DSD        DSINDEX    DSND
 DSNJ       DSNV       EADM       ENCLAVE    ENQ        ENQR
 HSM        IOQ        JES        JOB        LOCKSP     LOCKSU
 MSI        OPD        PCIE       PROC       PROCU      RG
 RLSDS      RLSLRU     RLSSC      SPACED     SPACEG     STOR
 STORC      STORCR     STORF      STORM      STORR      STORS
 SYSENQ     SYSINFO    SYSRG      SYSRTD     SYSSUM     SYSTREND
 SYSWKM     USAGE      WFEX       XCF        XCFGROUP   XCFOVW
 XCFPATH    XCFSYS     ZFSACT     ZFSFS      ZFSKN      ZFSOVW
 ZFSSUM

******************************* Bottom of data ********************************
```

Figure 28. Report Definition Initialization Panel ERB3RD1

The panel fields and their meanings are:

ACTION
Specifies the action you want RMF to perform as follows:

MODIFY - to change an existing RMF report
CREATE - to create a new report
DELETE - to delete an existing report

REPORT NAME
Specifies the name of the report that RMF is to modify, create, or delete. The report name must conform to ISPF naming conventions.

WLM MODE
Specifies the mode of the report, either compatibility or goal mode.

PROTOTYPE NAME
When you enter CREATE for ACTION, specify the name of an existing RMF report to use as a prototype or model for your report. RMF provides you those report values, which you can change when you modify or create your report.

When you enter MODIFY or DELETE for ACTION, you can ignore this field.

Phase driver information panel (ERB3RD2)

Press ENTER to display the next panel, the **Phase Driver Information** panel (ERB3RD2).

On this panel, you can specify the selection character(s) to use for the new or modified report on the Primary menu of a report session. You can also specify for each reporter phase the program or CLIST to modify, create, or print your report, or perform clean-up services and routines.

If you want to modify an existing RMF report without changing the layout or header information, you can provide your own program or CLIST for phase 2 on this panel. You can use ISPF services and commands like TBSORT, TBDELETE, or TBCREATE to perform these modifications during phase 2.

If you want to modify an existing RMF report format or layout without adding or deleting lines from a report, you can specify the name of the RMF report you want to modify for phase 1 (optionally for phase 2) and the name of the standard program that RMF uses to format RMF reports for phase 3. See PHASE 3 STRING in Figure 29 on page 119. You can then use the remaining report format definition utility panels to make the header and layout changes for the modified report.

If you want to create a report, you should use a prototype (see Figure 28 on page 118 for the Report Format Definition panel) and make sure to include the report selection on the Primary menu for the RMF report session.

Figure 29 on page 119 is an example of a Phase Driver Information panel that contains information about the SYSINFO report. It assumes that the new SYSCPU report will become available as option 4 in the User Selection menu.

```
ERB3RD2                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                        Section 1: Phase Driver Information
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

Select Strings format is: PGM(nnnnnnnn) PARM(mmm) or CMD(nnnnnnnn mmm)

SELECTION CHARACTERS  ===> U.4        Selection on Primary Option Panel

PHASE 1 SELECT STRING ===> PGM(ERB3RPH1) PARM(SYSINFO)
        TABLE  NAME   ===> ERBSYST3   Name of reporter phase 1 result table

PHASE 2 SELECT STRING ===>
        TABLE  NAME   ===> ERBSYST3   Optional name of phase 2 result table

PHASE 3 SELECT STRING ===> PGM(ERB3RDSP)

PHASE 4 SELECT STRING ===>
```

Figure 29. Phase Driver Information Panel (ERB3RD2)

The panel fields and their meanings are as follows:

SELECTION CHARACTERS
Specifies a 1 to 8 character alphanumeric value that RMF uses as a selection value on the Primary menu of a report session. You must have defined these selection characters in the menu panel.

If you enter a selection that is currently used on the Primary menu of a report session, RMF displays the report that you modify or create on this panel when you make the selection.

PHASE 1 SELECT STRING
Specifies the name of the program or CLIST that the reporter control module (ERB3RDPC) uses to generate the ISPF report table during phase 1. You must specify a CLIST for CMD or program for PGM. (Follow the rules for ISPF SELECT services.) If you are modifying an existing RMF report or creating a new report using a prototype, you must specify for PGM the program name ERBRPH1, and for PARM the command name of the RMF report that you are modifying or using as a prototype. If you are creating a new report, be sure to include the report as a selection on the Primary menu or on the User Selection menu.

See the RMF supplied phase driver table (ERBPHDS3) in Chapter 5, “Monitor III data reporter tables,” on page 143 for a list of the RMF program and PARM names.

PHASE 1 TABLE NAME
Specifies the name of the ISPF table that results when your program or CLIST is invoked during phase 1. You must specify this parameter if you have specified PHASE 1 SELECT STRING.

For a list of the RMF report data tables (PHDRTAB1) in the RMF supplied phase driver table (ERBPHDS3), see Chapter 5, “Monitor III data reporter tables,” on page 143.

PHASE 2 SELECT STRING
Specifies the name of the program or CLIST used to modify the ISPF report data table created in phase 1. If you are creating a new report without having specified a prototype, you must enter the name of your CLIST to create the new report. (Follow the rules for ISPF SELECT services.) If you are modifying only the report header or layout of an existing RMF report, you do not need to enter a PHASE 2 SELECT STRING.

PHASE 2 TABLE NAME
Specifies the name of the ISPF table that results after phase 2. If you have entered a value for PHASE 2 SELECT STRING, you must specify a valid phase 2 table name.

If you are modifying the report header or layout of an existing RMF report, you can enter the same name you entered for PHASE 1 TABLE NAME.

PHASE 3 SELECT STRING
Specifies the program or CLIST that RMF uses to initiate phase 3 to format your report.

If you do not provide a program or CLIST for this field, RMF skips the remaining report format definition utility panels and displays the report definition initialization panel ERB3RD1. When you invoke your report during an RMF session, RMF does not display the report.

If you are creating a report and you want RMF to display it, specify PGM(ERB3RDSP), the standard RMF display module.

PHASE 4 SELECT STRING
Specifies the program or CLIST that ERB3RDPC uses to initiate phase 4. This field is optional.

Report format information panel (ERB3RD3)

If you have entered a name for PHASE 3 SELECT STRING on ERB3RD2, RMF next displays the **Report Format Information** panel (ERB3RD3). This panel is the first in a series of panels that allows you to change the header and subheader layout of an RMF report.

On this panel (ERB3RD3), you can specify tabular or graphic, or both the tabular and graphic displays for the report, the panel name of the tabular version of the report, or specify the name of a report help panel.

Figure 30 on page 120 is an example of a Report Format Information panel for the SYSINFO report:

```
ERB3RD3                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                        Section 2: Report Format Information
WLM Mode:    GOAL

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.


REPORT MODE          ===> BOTH         TABULAR, GRAPHIC or BOTH


PANEL NAME           ===> ERB4CPU      Name of tabular report panel


HELP PANEL NAME      ===> ERB4CPU0     Name of HELP panel


LOGICAL LINE NUMBER  ===> SYSDTLLN     Name of table variable
SEQUENCE NUMBER      ===> SYSDTPSN     Name of table variable
```

Figure 30. Report Format Information Panel (ERB3RD3)

The panel fields and their meanings are as follows:

REPORT MODE
Specifies the display mode for the report. Valid values are as follows:

TABULAR
GRAPHIC
BOTH

PANEL NAME
Specifies the name of the ISPF display panel for the tabular version of the report when you enter TABULAR or BOTH for REPORT MODE.

For a tabular report, you must specify the name of the display panel that is to contain the report information. RMF-supplied panel names that you can use are ERB3DSI (if you are modifying or using the DI screen as a prototype), ERB3SRR (if you are modifying or using the STORR delay report as a prototype), ERB3SYS (if you are modifying or using the SYSINFO report as a prototype), ERB3WFX (if you are modifying or using the WFEX report as a prototype), or ERB3CMN (if you are modifying or using any other report as a prototype).

If you specify the name of your own panel, make sure that the panel includes the following information:

- Output fields for 2 standard header lines (DSPHDR1 and DSPHDR2)
- Output fields for up to 5 subheader lines (DSPSUBH1 - DSPSUBH5) contained in the RMF report you want to modify
- Output fields for up to 3 column header lines (FMTCOLH1 -FMTCOLH3) contained in the RMF report you want to modify. For a description of the report format table ERBFMTS3, see Chapter 5, “Monitor III data reporter tables,” on page 143.
- Up to 3 model line variables (FMTMODL1 - FMTMODL3) contained in the model section of the RMF report you want to modify. For a description of the entries in the report format table ERBFMTS3, see Chapter 5, “Monitor III data reporter tables,” on page 143.
- The command line (defined by variable ZCMD) and scroll amount field (defined by variable AMT)

Also, ensure that the user-defined panel for your report includes an initialization (INIT), reinitialization (REINIT), and processing (PROC) section as in the RMF-supplied panels.

If you enter GRAPHIC for REPORT MODE, leave PANEL NAME blank.

HELP PANEL NAME
When you enter a value for PANEL NAME, specifies the name of the ISPF help panel that contains help information for your report. The field is optional.

LOGICAL LINE NUMBER/SEQUENCE NUMBER
Specifies the name of key variables in the data table of the RMF report you are modifying. A logical line number identifies a logical group of related data rows within a report; a line sequence number identifies each physical table row that belongs to the logical group.

The logical line number (that identifies the entire data group) is 1; the sequence number (the number of physical lines that belong to the logical group and include the volume serial/device type on one line and the space type on the second line of the graphic report) is 2 or more.

When you toggle between tabular and graphic reports, RMF uses these variables to synchronize the line or bar displayed on the screen (the beginning of a logical group of data table rows). For examples of RMF report data tables, see Chapter 5, “Monitor III data reporter tables,” on page 143.

Report header layout panels (ERB3RD4 and ERB3RD5)

Press ENTER to display the next panel, ERB3RD4, the **Report Header Layout** panel.

Each RMF report contains report headings, subheadings, and columns that you can modify. The Report Header Layout panels (ERB3RD4 and ERB3RD5) allow you to change up to 2 header lines for the tabular and graphic versions of the report.

On the first of these panels (ERB3RD4), you can specify the header lines and header variables for your report. At the bottom of the panel, enter the header lines exactly as you want them to appear on your report. You can use the variables listed on the panel to appear in the headings of your report. (Panel ERB3RD4 lists variables from header data table ERBHDRS3. For the meaning of all variables in ERBHDRS3, see Chapter 5, “Monitor III data reporter tables,” on page 143.)

If a variable name is too long to enter in the header line, you can use a placeholder (&Z). After you press ENTER, you define these placeholders with variable names on the next panel.

Figure 31 on page 122 is an example of a report format definition panel ERB3RD4 that shows you the headings and variables for the SYSINFO report with the modified report title `CPU Information`:

```
ERB3RD4                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL          Section 3: Report Header Layout

Enter or change the report header lines.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).
If you specify variable placeholders (&Z) the next panel will ask you
to specify the variable name that is to replace each &Z

The following variables are available for use in the header:
  &ERBSID     &ERBSAMPL   &ERBTIME    &ERBRMFVD   &ERBSNUM
  &ERBHCTXT   &ERBDATE    &ERBRANGE   &ERBSPXID   &ERBSAMWL

Variables ERBSID, ERBDATE, ERBTIME and ERBRANGE will be supported as input
fields only, if they are part of second header line.

Enter or change up to two report heading lines:

           &ERBHCTXT    &ERBRMFVD  CPU Information
Samples: &Z      System: &Z    Date: &ERBDATE  Time: &ERBTIME  Range: &Z    Sec
```

Figure 31. Report Header Layout Panel (ERB3RD4)

In Figure 31 on page 122, two report header lines appear at the bottom of the panel and ten variable names are available for the header lines.

- Variables &ERBHCTXT and &ERBRMFVD are specified at the beginning of the first header line.
- Variables &ERBDATE and &ERBTIME are specified for `Date` and `Time`.
- Placeholders (&Z) for the other variables (&ERBSID for session id, &ERBSAMPL for samples, and &ERBRANGE for range) appear in the appropriate fields of the header lines and indicate that the variable names they represent might not fit in the space provided. These placeholders can be defined on the next panel.

Press ENTER to display the second **Report Header Layout** panel (ERB3RD5).

On ERB3RD5, you can specify variable names for any Z placeholders you have used. The headings, variables names, and placeholders as you entered them on ERB3RD4 appear at the top of the panel. The variable names appear under the headings in the order specified on ERB3RD4. You can specify your own variable names in the spaces provided; however, in order for RMF to display the user-specified variables during a report session, they must be in the function pool for phase 3 or in the shared ISPF variable pool. Otherwise, blanks appear in the report. See “Installing your own phases” on page 134.

You must specify a number for each Z placeholder and its corresponding variable. Numbers must start with 1 and continue in sequence. There must be a one-to-one correspondence between placeholders and variable names, each pair with a unique number assigned to indicate the order of placement of the variable.

Figure 32 on page 123 is an example of Report Header Layout panel ERB3RD5 that defines the placeholders used on the previous panel. If you do not have placeholders to define, press ENTER to get the next panel.

```
ERB3RD5                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                              Section 3: Report Header Layout
WLM Mode:    GOAL

The following report header lines have been specified:
           &ERBHCTXT    &ERBRMFVD  CPU Information
Samples: Z1      System: Z2    Date: &ERBDATE  Time: &ERBTIME  Range: Z3    Sec

Specify the placeholder (Z) number next to the variable name to replace each Z
above.To continue press ENTER.To go backwards press END.To quit enter CANCEL.


     &ERBSID   ===> 2            &ERBHCTXT ===> __           &ERBSAMPL ===> 1
     &ERBDATE  ===> __           &ERBTIME  ===> __           &ERBRANGE ===> 3
     &ERBRMFVD ===> __           &ERBSPXID ===> __           &ERBSNUM  ===> __
     &ERBSAMWL ===> __      ===> _________ ===> __      ===> _________ ===> __
===> _________ ===> __      ===> _________ ===> __      ===> _________ ===> __
```

Figure 32. Report Header Layout Panel (ERB3RD5)

- Variable &ERBSAMPL that contains the number of samples replaces Z1.
- Variable &ERBSID that contains the session id replaces Z2 in the first header line of the report.
- &ERBRANGE that contains the range value replaces Z3 in the second header line.

Depending on your selection on panel ERB3RD3, you will continue as follows:

- If you specified TABULAR or BOTH for report mode, RMF displays the **Report Subheader Layout** panel ERB3RD6.
- If you specified GRAPHIC for report mode, RMF displays the **Graphic Parameter Definition** panel ERB3RDB, see “Graphic parameter definition panels (ERB3RDB, ERB3RDC, ERB3RDD)” on page 127.

Report subheader layout panels (ERB3RD6 and ERB3RD7)

The Report Subheader Layout panel (ERB3RD6) displays up to five subheader lines of an existing RMF report. (Subheader lines are any lines in an RMF report that appear between the two standard header lines and the column headings.) ERB3RD6 also lists the variables that are available for use in the subheader lines of the modified report.

At the bottom of ERB3RD6, you enter the subheader lines exactly as you want them to appear on your report. You can use the variables listed on the panel to appear in the subheadings of your report. Panel ERB3RD6 lists variables from header data table ERBHDRS3.

If a variable name is too long to appear in the header line, you can use a placeholder (&Z). After you press ENTER, you define these placeholders with variable names on the next panel.

Figure 33 on page 124 is an example of a Report Subheader Layout panel ERB3RD6 that shows the subheadings of the SYSINFO report.

```
ERB3RD6                RMF Report Format Definition
Command ===>

Report Name: SYSINFO    WLM Mode: GOAL       Section 4: Report Subheader Layout

Enter or change the report subheader lines.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).
If you specify variable placeholders (&Z) the next panel will ask you
to specify the variable name that is to replace each &Z.

The following variables are available for use in the subheader:
  &SYSPARVC    &SYSMODVC    &SYSMDLVC    &SYSTSVVC     &SYSIPVVC    &SYSPOLVC
  &SYSVEPVC    &SYSPRVVC    &SYSCUVVC    &SYSTSEVC     &SYSOPVVC    &SYSPADVC
  &SYSPRIVC    &SYSICVVC    &SYSLCPVC    &SYSAPIVC     &SYSPATVC    &SYSPRTVC
  &SYSAPTVC    &SYSAICVC    &SYSATCVC    &SYSLOAVG     &SYSTCTVC    &SYSUTCVC
Enter or change up to five report subheading lines:
&Z                  &Z   Model &Z           Appl%:     &Z   Policy: &Z
CPs Online:  &Z       Avg CPU Util%:  &Z      EAppl%:    &Z   Date:   &Z
AAPs Online: &Z    &Z                       Appl% AAP: &Z   Time:   &Z
IIPs Online: &Z                             Appl% IIP: &Z
```

Figure 33. Report Subheader Layout Panel (ERB3RD6)

In Figure 33 on page 124, subheader lines appear at the bottom of the panel and 24 variable names from the SYSINFO report are available. You can modify these subheader lines and indicate where you want the available variables to appear in them.

Press ENTER to display the next panel ERB3RD7, the second Report Subheader Layout panel.

On this panel, you can specify variable names for any Z placeholders you have used. For a description of how to replace placeholders with variable names, see the Report Header Layout panel (Figure 32 on page 123).

Figure 34 on page 124 shows panel ERB3RD7 that defines placeholders used on the previous panel.

```
ERB3RD7                RMF Report Format Definition
Command ===>

Report Name: SYSINFO    WLM Mode: GOAL       Section 4: Report Subheader Layout

The following report subheader lines have been specified:
Z1                  Z2   Model Z3           Appl%:     Z4   Policy: Z5
CPs Online:  Z6       Avg CPU Util%:  Z7      EAppl%:    Z8   Date:   Z9
AAPs Online: Z10   Z11                       Appl% AAP: Z12  Time:   Z13
IIPs Online: Z14                             Appl% IIP: Z15


Specify the placeholder (Z) number next to the variable name to replace each Z
above. To continue press ENTER. To go backwards press END.To quit enter CANCEL.

     &SYSPARVC ===> 1            &SYSMODVC ===> 2            &SYSMDLVC ===> 3
     &SYSTSVVC ===> 4            &SYSIPVVC ===> __           &SYSPOLVC ===> 5
     &SYSVEPVC ===> __           &SYSPRVVC ===> 6            &SYSCUVVC ===> 7
     &SYSTSEVC ===> 8            &SYSOPVVC ===> __           &SYSPADVC ===> 9
     &SYSPRIVC ===> 10           &SYSICVVC ===> __           &SYSLCPVC ===> 11
     &SYSAPIVC ===> 12           &SYSPATVC ===> 13           &SYSPRTVC ===> 14
     &SYSAPTVC ===> 15           &SYSAICVC ===> __           &SYSATCVC ===> __
     &SYSLOAVG ===> __           &SYSTCTVC ===> __           &SYSUTCVC ===> __
```

Figure 34. Report Subheader Layout Panel (fERB3RD7)

Report column layout panels (ERB3RD8 and ERB3RD9)

Press ENTER to display the next panel, ERB3RD8, the report subheader first **Report Column Layout** panel.

On this panel, you can modify report columns. You can enter up to three column header lines as you want them to appear in the report.

You can specify up to three model lines for your columns by using an attribute character followed by a variable name or placeholder (&Z). (See DATA ATTRIBUTE CHARACTERS described below.)

You can use the variable names listed at the bottom of the panel to appear in the columns of your report. This panel also allows you to specify a placeholder (&Z) for any variable name you want to use. (Panel ERB3RD8 lists variables from the data table of the RMF report you are modifying. All variables might not appear on the first page of the panel. Scroll through the panel and select the variable names you need. For information about RMF report data tables, see Chapter 5, “Monitor III data reporter tables,” on page 143.) You can define placeholders for variable names on the next panel.

Figure 35 on page 125 is an example of ERB3RD8 that shows report column headings for the modified SYSINFO report with columns that contain data about TCB%, SRB%, and execution velocity. The details about delay percentages have been removed.

```
ERB3RD8                 RMF Report Format Definition              Line 1 of 10
Command ===>                                                   Scroll ===> PAGE

Report Name: SYSINFO                            Section 5: Report Column Layout
WLM Mode:    GOAL

Enter or change the following information. To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

DATA ATTRIBUTE CHARACTERS ===> _?|    Define meaning in attribute section
                                     of associated table display (ERB3SYS).

Enter or change up to three column header lines:
Group     T WFL --Users--  RESP TRANS  CPU   TCB   SRB  -AVG USG-   -Avg Del-
             %   TOT  ACT  Time  /SEC   %     %     %   PROC  DEV   PROC  DEV
-------------------------------------------------------------------------------

Enter or change up to three model lines:
?Z        ?Z?Z  ?Z   ?Z    ?Z   ?Z    ?Z    ?Z    ?Z    ?Z   ?Z     ?Z   ?Z


--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

The following variables are available for use in the model lines:
  SYSNAMVC     SYSTYPVC     SYSWFLVC     SYSTUSVC     SYSAUSVC     SYSTRSVC
  SYSAFCVC     SYSVECVC     SYSAUPVC     SYSAUDVC     SYSADPVC     SYSADDVC
  SYSADSVC     SYSADUVC     SYSADOVC     SYSADEVC     SYSADJVC     SYSADHVC
  SYSADXVC     SYSADNVC     SYSADMVC     SYSCPUVC     SYSSRBVC     SYSTCBVC
  SYSIFAVC     SYSCPVC      SYSIFCVC     SYSRSPVC     SYSVELVC     SYSUGMVC
  SYSUGPVC     SYSUGDVC     SYSWGDVC     SYSWGPVC     SYSDGMVC     SYSUJMVC
  SYSDJMVC     SYSDGEVC     SYSDGHVC     SYSDGDVC     SYSDGJVC     SYSDGOVC
  SYSDDSIP     SYSEAPVC     SYSLPIVC     SYSSUPVC     SYSSUCVC     SYSDTLLN
```

Figure 35. Report Column Layout Panel (ERB3RD8)

DATA ATTRIBUTE CHARACTERS

Specifies the ISPF characters used to indicate the start of a data field. Specify the data attribute characters before each variable name or placeholder (&Z) used in the model lines.

You must specify the name of a panel for the tabular version of a new or modified report. For RMF-supplied panels, the attribute characters appear as follows:

- a question mark (?) indicates that the output display characters appear unhighlighted (low intensity) in turquoise
- a slash (|) indicates that the output display characters appear highlighted (high intensity) in white
- a blank indicates that the input display characters appear unhighlighted (high intensity) in green

For user-defined panels, be sure that the data attribute characters match the characters in the attribute section of your ISPF display panel. See PANEL NAME on the report format information panel (ERB3RD3).

Press ENTER to display the next panel ERB3RD9, the second Report Column Layout panel.

On this panel, you can specify variable names for any Z placeholders you have used. The variable names available on the previous panel are listed at the bottom; you can add your own variable names in the spaces provided. If your variable names are not available when you invoke the report, blanks will appear instead of data. See the report header information panel (ERB3RD5) in Figure 32 on page 123 for a description of how to replace placeholders with variable names.

If not all variable names appear on the first page of the panel, scroll through the remaining pages of the panel to see all available variable names.

Figure 36 on page 126 is an example of Report Column Layout panel ERB3RD9 that defines placeholders used on the previous panel.

```
ERB3RD9                 RMF Report Format Definition             Line 1 of 21
Command ===>                                                  Scroll ===> PAGE

Report Name: SYSINFO                           Section 5: Report Column Layout
WLM Mode:    GOAL

The following report column header and model lines have been specified:
Group     T WFL --Users--  RESP TRANS  CPU   TCB   SRB  -AVG USG-   -Avg Del-
            %   TOT  ACT  Time  /SEC   %     %     %   PROC  DEV   PROC  DEV

Z1        Z2Z3  Z4   Z5    Z6   Z7    Z8    Z9    Z10   Z11  Z12    Z13  Z14



Specify the placeholder (Z) number next to the variable name to replace each Z
above. To continue press ENTER. To go backwards press END.To quit enter CANCEL.

     SYSNAMVC ===> 1             SYSTYPVC ===> 2             SYSWFLVC ===> 3
     SYSTUSVC ===> 4             SYSAUSVC ===> 5             SYSRSPVC ===> 6
     SYSTRSVC ===> 7             SYSCPUVC ===> 8             SYSTCBVC ===> 9
     SYSSRBVC ===> 10            SYSAUPVC ===> 11            SYSAUDVC ===> 12
     SYSADPVC ===> 13            SYSADDVC ===> 14            SYSAFCVC ===> __
     SYSVECVC ===> __            SYSADSVC ===> __            SYSADUVC ===> __
     SYSADOVC ===> __            SYSADEVC ===> __            SYSADJVC ===> __
     SYSADHVC ===> __            SYSADXVC ===> __            SYSADNVC ===> __
     SYSADMVC ===> __            SYSIFAVC ===> __            SYSCPVC  ===> __
     SYSIFCVC ===> __            SYSVELVC ===> __            SYSUGMVC ===> __
     SYSUGPVC ===> __            SYSUGDVC ===> __            SYSWGDVC ===> __
     SYSWGPVC ===> __            SYSDGMVC ===> __            SYSUJMVC ===> __
     SYSDJMVC ===> __            SYSDGEVC ===> __            SYSDGHVC ===> __
     SYSDGDVC ===> __            SYSDGJVC ===> __            SYSDGOVC ===> __
     SYSDGPVC ===> __            SYSDGSVC ===> __            SYSDGUVC ===> __
```

Figure 36. Report Column Layout Panel (ERB3RD9)

Command line layout panel (ERB3RDA)

Press ENTER to display the next panel ERB3RDA, the **Command Line Layout** panel.

On this panel, you can specify the format of the command line and scroll line as you want them to appear on the hardcopy of the tabular report. You must also define the command line and scroll line on the display panel of the tabular report.

Figure 37 on page 127 is an example of Command Line Layout panel ERB3RDA.

```
ERB3RDA                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU     WLM Mode: GOAL       Section 6: Command Line Layout


Enter or change the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

You may intermix: text, variables, and variable placeholders (&Z).

The following variables are available for use in the command line:
  &ZCMD          &AMT

Enter or change the command line:

Command ===>

Specify a variable name in each of the entry fields to replace each Z above.

Z1 ===>
Z2 ===>
Z3 ===>
```

Figure 37. Command Line Layout Panel (ERB3RDA)

Graphic parameter definition panels (ERB3RDB, ERB3RDC, ERB3RDD)

If you specified BOTH or GRAPHIC for report mode on ERB3RD3, RMF displays the first **Graphic Parameter Definition** panel, ERB3RDB.

On this panel, you can specify general information about the graphic version of the report.

Note: If you specified TABULAR for report mode on the report format information panel (ERB3RD1) or used DI or WFEX as a prototype, the report format definition utility displays panel ERB3RDF. This panel allows you to save your changes and view the tabular report you have created or cancel your changes. See “Saving or cancelling changes on panel ERB3RDF” on page 130.

Figure 38 on page 127 is an example of the Graphic Parameter Definition panel ERB3RDB that specifies general information for the graphic version of the SYSINFO report:

```
ERB3RDB                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                     Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.


                               GENERAL INFORMATION



NAME FOR HELP PANEL ===> ERBGSYS0     Name of HELP PANEL, if any

TITLE FOR Y-AXIS    ===> Average Number of Active Users

MINIMUM AXIS RANGE  ===> 1            Axis will contain at least this
                                      number of data points
SELECTION RULE      ===> 1            Specify 0, 1, 2 or 3
```

Figure 38. Graphic Parameter Definition Panel (ERB3RDB)

The fields and their meanings follow:

NAME FOR HELP PANEL
Specifies the name of the help panel that you provide for the graphic report. The field is optional.

TITLE FOR Y-AXIS
Specifies a line of text (maximum of 50 characters) to appear as a label for the bar graph in the graphic version of the report. Sample lines that appear in the graphic parameter table (ERBPTGS3) are:

- Percentage of Each User's Time
- Percentage of the User's Time
- Average Number of Active Users

For an example of the graphic parameter table (ERBPTGS3), see Chapter 5, “Monitor III data reporter tables,” on page 143.

MINIMUM AXIS RANGE
Specifies the length of the bar graph depending on the text specified in TITLE FOR Y-AXIS as follows. For each line of text listed in the previous example, the minimum axis range is as follows:

- 100 for “Percent of Each User's Time”
- 100 for “Percent of the User's Time”
- 1 for “Average Number of Active Users”

If the length of the largest bar in the report exceeds the value you specify, RMF uses the length of the largest bar.

For an example of the graphic parameter table (ERBPTGS3), see Chapter 5, “Monitor III data reporter tables,” on page 143.

SELECTION RULE
Specifies how the lines of the tabular report appear as bar graphs on the graphic version of the report. You can select one of the following values:

- 0 - One bar corresponds to one line of the RMF tabular report
- 1 - One bar corresponds to one line of the RMF tabular report with sequence number 1 (for example, DEV, HSM, JES, STOR, PROC, DELAY, SYSINFO, and ENQ)
- 2 - One bar corresponds to the summary of logical lines of the report (for example, ENQR, DEVR reports)
- 3 - Two bar types can result from all logical lines of a logical block in the RMF tabular report (for example, STORR report) as follows:
  - Bar type 1 corresponds to a line of the tabular report with sequence number 1
  - Bar type 2 corresponds to each additional line of the logical block for a tabular report with a sequence number greater than 1

For an example of the graphic parameter table (ERBPTGS3), see Chapter 5, “Monitor III data reporter tables,” on page 143. For a description of logical line number and sequence number, see the panel field description for ERB3RD3 (Figure 30 on page 120).

Press ENTER to display the next panel, ERB3RDC, the second **Graphic Parameter Definition** panel.

On this panel, you can specify labels for the graphic bars in the report. You can specify variable names for bar type 1 labels and bar type 2 labels.

Figure 39 on page 129 is an example of the Graphic Parameter Definition panel ERB3RDC.

```
ERB3RDC                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                     Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information.  To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

                         LABEL INFORMATION FOR BAR TYPE I

PRIMARY LABEL       ===> SYSNAMVC  Variable name containing label
SECONDARY LABEL     ===> ________  Variable name containing label
PRIMARY COMPOSITE   ===> _____     Prefix of label
SECONDARY COMPOSITE ===> _____     Prefix of label

                        LABEL INFORMATION FOR BAR TYPE II

PRIMARY LABEL       ===> ________  Variable name containing label
SECONDARY LABEL     ===> ________  Variable name containing label
PRIMARY COMPOSITE   ===> _____     Prefix of label
SECONDARY COMPOSITE ===> _____     Prefix of label
```

Figure 39. Graphic Parameter Definition Panel (ERB3RDC)

The panel fields and their meanings are as follows:

PRIMARY LABEL/SECONDARY LABEL
Specifies an 8 character variable name for a data value in the graphic version of the report. You can use the variable names that appear in the ISPF data table of the corresponding tabular report.

For example, in Figure 39 on page 129, the Primary label will appear as average number of active users (SYSNAMVC) on the graphic version of the SYSCPU report.

See Chapter 5, “Monitor III data reporter tables,” on page 143 for examples of the Graphic Parameter table (ERBPTGS3) and the RMF Report Data tables.

PRIMARY COMPOSITE/SECONDARY COMPOSITE
Specifies up to 5 characters of text as a prefix to the variable label specified in PRIMARY/SECONDARY LABEL. In Figure 39 on page 129, no composite labels appear in the SYSCPU report. You can specify a prefix to appear in the graphic version of the report. The prefix is concatenated to the rightmost contents of the report table variable specified in PRIMARY/SECONDARY label.

See Chapter 5, “Monitor III data reporter tables,” on page 143 for examples of the Graphic Parameter table (ERBPTGS3) and the RMF Report Data tables.

BAR TYPE refers to the number of bars used in the report depending on the logical line and sequence numbers.

See LOGICAL LINE/SEQUENCE NUMBER in the field descriptions for ERB3RD3 (Figure 30 on page 120).

Press ENTER to display the next panel, ERB3RDD, the third **Graphic Parameter Definition** panel.

On this panel, you can specify data columns that you want to appear in the graphic version of the report.

Figure 40 on page 130 is an example of the Graphic Parameter Definition panel ERB3RDD.

```
ERB3RDD                 RMF Report Format Definition
Command ===>

Report Name: SYSCPU                     Section 7: Graphic Parameter Definition
WLM Mode:    GOAL
             Definitions on this panel are independent of WLM mode.

Enter the following information. To continue press ENTER.
To quit enter CANCEL. To go backwards press END.

                   COLUMN SPECIFICATION FOR GRAPHIC BAR TYPES

         NAME               LEGEND ID           TRANS ID           BAR TYPE ID

 1. ===> SYSADPVC       ===> 14             ===> 0             ===> 1
 2. ===> SYSADDVC       ===> 08             ===> 0             ===> 1
 3. ===> SYSADSVC       ===> 15             ===> 0             ===> 1
 4. ===> SYSADUVC       ===> 28             ===> 0             ===> 1
 5. ===> SYSADOVC       ===> 29             ===> 0             ===> 1
 6. ===> SYSADEVC       ===> 09             ===> 0             ===> 1
 7. ===> SYSAUPVC       ===> 19             ===> 0             ===> 1
 8. ===> SYSAUDVC       ===> 18             ===> 0             ===> 1
 9. ===> ________       ===> __             ===> _             ===> _
10. ===> ________       ===> __             ===> _             ===> _
```

Figure 40. Graphic Parameter Definition Panel (ERB3RDD)

The panel fields and their meanings are as follows:

NAME
Specifies an 8 character variable name for a data value from the corresponding tabular report. This value will appear as a bar column in the graphic version of the report. The bar column can be a single bar (bar type 1) or a stacked bar (bar type 2) depending on what you specify for BARTYPE ID. See Chapter 5, “Monitor III data reporter tables,” on page 143 for examples of RMF report data tables.

LEGEND ID
Specifies a number that corresponds to the color, pattern and the text of the graphic chart legend. Variables specified for NAME will appear in the color specified for LEGEND ID. You can specify a decimal value from 04 to 27; the numbers must match the color ID entries on the Color Graphic Option panels.

TRANS ID
Specifies a number that controls how the values for the variable in NAME are scaled on the bar graph in the graphic version of the report.

- 0 - value appears as is; no division is performed
- n - value is divided by 10^n where **n** equals an integer from 1 to 9.

See Chapter 5, “Monitor III data reporter tables,” on page 143 for examples of RMF report data tables.

BARTYPE ID
Specifies a value that indicates where the data value for the variable in NAME appears for bar types in the graphic version of the report:

- 0 - indicates the value appears in both bar types
- 1 - indicates the value occurs in bar type 1
- 2 - indicates the value occurs in bar type 2

If you specified label information for only bar type 1 on the report parameter definition panel (ERB3RDC), you must specify bar type 1.

Saving or cancelling changes on panel ERB3RDF

Once you have created or modified a report using the report format definition utility panels, RMF displays panel ERB3RDF, which allows you to confirm or cancel your changes.

```
ERB3RDF                 RMF Report Format Definition
Command ===>


This is a confirmation/cancellation panel for report: SYSCPU
                              related with WLM mode: GOAL



The following actions are allowed:


       Type  SAVE   command to save report

       Type  CANCEL command to cancel processing

       Press END    key to go one step backwards

       Press ENTER  key to see the sample report
```

Figure 41. Configuration/Cancellation Panel (ERB3RDF)

You can get a report with sample data just to verify the correct layout of the report. In this example, some values are not displayed because they are not part of the sample data.

```
ERB4CPU                 RMF V2R4  CPU Information               Line 1 of 20
Command ===>                                                  Scroll ===> PAGE

Samples:     60  System: SYSE  Date: 06/08/2019  Time: 08.59.00  Range:     60Sec

Partition:    SYSE    2084  Model 314         Appl% :     22  Policy: STANDARD
CPs Online:   4.0     Avg CPU Util%:   24     EAppl%:     22  Date:   05/12/05
AAPs Online:  1.0     Avg MVS Util%:   26     Appl% AAP:  19  Time:   12.20.27
IIPs Online:    -                            Appl% IIP:   -

Group     T WFL --Users--  RESP TRANS  CPU   TCB   SRB  -AVG USG-   -Avg Del-
             %   TOT  ACT  Time  /SEC   %     %     %   PROC  DEV   PROC  DEV

*SYSTEM      98  167    2        0.12  21.6  21.5   0.2  1.6  0.1    0.0  0.0
*TSO            3    0        0.12   0.0   0.0   0.0  0.0  0.0    0.0  0.0
*BATCH       99    3    2        0.00  20.5  20.5   0.0  1.5  0.0    0.0  0.0
*STC         91  158    0        0.00   1.2   1.0   0.2  0.1  0.1    0.0  0.0
*ASCH           0    0        0.00   0.0   0.0   0.0  0.0  0.0    0.0  0.0
*OMVS           3    0        0.00   0.0   0.0   0.0  0.0  0.0    0.0  0.0
*ENCLAVE        0  N/A         N/A   N/A   N/A   N/A  0.0  N/A    0.0  N/A
BATCH     W  99    4    2  .000  0.00  20.5  20.5   0.0  1.5  0.0    0.0  0.0
BATCHLOW  S  99    3    2  .000  0.00  20.5  20.5   0.0  1.5  0.0    0.0  0.0
OMVSKERN  S        1    0  .000  0.00   0.0   0.0   0.0  0.0  0.0    0.0  0.0
OMVS      W        2    0  .000  0.00   0.0   0.0   0.0  0.0  0.0    0.0  0.0
OE        S        2    0  .000  0.00   0.0   0.0   0.0  0.0  0.0    0.0  0.0
STC       W 100   26    0  .000  0.00   0.1   0.1   0.0  0.0  0.0    0.0  0.0
STCDEF    S 100   26    0  .000  0.00   0.1   0.1   0.0  0.0  0.0    0.0  0.0
SYSTEM    W  90  132    0  .000  0.00   1.0   0.9   0.2  0.1  0.1    0.0  0.0
SYSSTC    S 100  111    0  .000  0.00   0.3   0.3   0.0  0.0  0.0    0.0  0.0
SYSTEM    S  88   21    0  .000  0.00   0.7   0.6   0.2  0.1  0.1    0.0  0.0
TSO       W        3    0  .010  0.17   0.0   0.0   0.0  0.0  0.0    0.0  0.0
TSODEF    S        3    0  .010  0.17   0.0   0.0   0.0  0.0  0.0    0.0  0.0
BCP       R  99    3    2  .000  0.00  20.5  20.5   0.0  1.5  0.0    0.0  0.0
```

Figure 42. Initial Version of the SYSCPU Report

The report shows that adjustments for some columns are necessary. You can do this either by stepping back to panel ERB3RD8 before you save the report or by modifying the stored report.

Enter SAVE to save the report or CANCEL to cancel your changes and return to the report definition initialization panel (ERB3RD1). If you save the report, RMF redisplays panel ERB3RD1 with a message that tells you the report has been modified or created. To exit the sample report panel and return to panel ERB3RDF, press END.

Deleting a user-defined report

If you specify DELETE for a report on the report format definition panel, RMF displays panel ERB3RDE. To confirm the deletion of the report, press ENTER and the report is deleted. To cancel the deletion, type CANCEL and press ENTER. RMF returns you to ERB3RD1.

Note: You can only delete a user-defined report. RMF does not allow you to delete an existing RMF report.

Ending the report utility

You can end the report format definition utility session by pressing END (PF3) on the report format definition panel (ERB3RD1) or by specifying CANCEL on any panel.

Implementing the report

To make the new SYSCPU report available, it needs to be integrated in a Monitor III selection panel. As defined initially, the report shall be added to the User Selection menu ERB3USR. You can do this by these modifications to the definition of the panel:

```
)attr default(!+_)
/*********************************************************************
/*          PANEL NAME: ERB3USR
/*

…

)body expand("") cmd(zcmd)
+          !         RMF User-written Report Selection Menu      " "
!Selection ===>_ZCMD " " +
+
<Enter selection number or command for desired report.
+
+
!  2<DSD            +Detailed Storage Delays
!
!  4<SYSCPU         +Modified SYSINFO including CPU details
+

…
```

Figure 43. Modifications in User Selection Menu Definition (ERB3USR) - Part 1

```
 /* translate subsystem selections ********************************** */
    &erbcmdc = trans(&erbcmdc
                  2,'DSD'
                  4,'SYSCPU'
                 ST,'SYSTREND'
                 DA,'DEVN'
                 DT,'DEVT'
                  *,*)
 /* Checks if command input is a valid RMF command.                  */
    ver(&erbcmdc,LIST, CANCEL, FIND, GRAPHIC, ICU, HARDCOPY, RESET,
        RFIND, TABULAR, TOGGLE,
        MSI, DSD, RG, SYSCPU,
        DEVN, DEVT, SYSTREND,
        MSG=ERB562I)
 /* Checks if command input is a valid on this screen.               */
    ver(&erbcmdc,LIST,;
        MSI, DSD, RG, SYSCPU,
        DEVN, DEVT, SYSTREND,
        MSG=ERB573I)

…

 /* selects action according to entered input ********************** */
    &zsel = trans(&erbcmdc
              MSI,'PGM(ERB3RDPC) PARM(MSI)'
              DSD,'PGM(ERB3RDPC) PARM(DSD)'
               RG,'PGM(ERB3RDPC) PARM(RG)'
           SYSCPU,'PGM(ERB3RDPC) PARM(SYSCPU)'
             DEVN,'PGM(ERB3RDPC) PARM(DEVN)'
             DEVT,'PGM(ERB3RDPC) PARM(DEVT)'
         SYSTREND,'PGM(ERB3RDPC) PARM(SYSTREND)'
                *,' ')
)END
```

Figure 44. Modifications in User Selection Menu Definition (ERB3USR) - Part 2

If you call the User Selection menu, you now get this new version:

```
ERB3USR             RMF User-written Report Selection Menu
Selection ===>

Enter selection number or command for desired report.



  2 DSD             Detailed Storage Delays

  4 SYSCPU          Modified SYSINFO including CPU details

Device Reports
 DA DEVN            Device Activity
 DT DEVT            Device Trend
                    Device   => _______

System Reports
 ST SYSTREND        System and Workload Trend
                    Workload => ________
```

Figure 45. Modified User Selection Menu (ERB3USR)

Special considerations for modifying reports

If you want to add or delete lines in an existing RMF report or sort lines of a report without modifying the report heading, consider the following when you use the report format definition utility:

- Each RMF report data table (PHDRTAB1 in the phase driver table) contains the ISPF key type variables for the logical line number and line sequence number for the report. Each data table lists the logical lines and the sequence number(s) for logical lines of data in the report in ascending order. Sequence numbers for each logical line begin with 1. When you add, delete, or sort lines of an RMF report, be sure that the output table of your report (PHASE 1 or 2 TABLE on phase driver information panel ERB3RD2) arranges logical line and sequence numbers in ascending order.
- If you delete a line of a report with sequence number 1, you must also delete the logical line number of the report from the data table.

- If you want to rearrange the lines of an RMF report, you can use the ISPF service TBSORT as part of the CLIST you specify for phase 2. You can specify the CLIST with TBSORT on the phase driver information panel (ERB3RD2) as follows:

  ```
  CMD(mysort)
  ```

  where “mysort” is the name of your CLIST.

Installing your own phases

When you select a report during a reporter session, RMF uses ISPF SELECT services to generate report data tables and display the reports. You can supply your own routines for any of the 4 phases to produce user-defined reports. See “Data reporter phases” on page 113 for a description of the phases RMF invokes.

The following ISPF shared variables are available during all phases. They can be updated in Phase 1 by the Data Retrieval Service.

ERBDATE,ERBTIME
The ISPF shared pool variables that contain the beginning date and time of the display data.

ERBRANGE
The ISPF shared pool variable that contains the range time of the display data. The beginning date/time plus the range time of the display data equals the ending date/time.

ERBSID
The ISPF shared pool variable that contains the id of the system on which the data was collected.

ERBSAMPL
The ISPF shared pool variable that contains the number of data samples for the time range.

ERBRMFVD
The ISPF shared pool variable that contains the RMF version number of the data gatherer which collected the data. The format is **RMF VvRr** (that is, RMF V2R4).

ERBDTBEG
The ISPF shared pool variable which contains the beginning date/time value for the requested time range. The format is **MMDDYYYYHHMMSS** (that is, 09252019183050 represents September 25, 2019 at 18:30:50).

ERBDTEND
The ISPF shared pool variable which contains the ending date/time value for the requested time range. The format is **MMDDYYYYHHMMSS** (that is, 08272019173010 represents August 27, 2019 at 17:30:10).

ERBMNTIM
The ISPF shared pool variable which contains the Monitor III data gatherer MINTIME option value that was in effect when the data was gathered. The data is in external decimal format.

Phase 1

If you want to use your own program for phase 1, you must ensure that the ISPF shared pool variable PHDRPH1 contains the name of your program or CLIST. This variable appears in the phase driver table (ERBPHDS3) as an ISPF SELECT string. For RMF reports, the PARM value of the string matches the name of the RMF report command. You can use the report definition format utility to specify your own PHASE 1 SELECT STRING. See Chapter 5, “Monitor III data reporter tables,” on page 143 for an example of the phase driver table (ERBPHDS3) entries and how they are specified.

If you want to change the time range from which your data is collected, you can invoke the Data Retrieval Service (ERB3RDRS) module from your phase 1 program. See “Data retrieval service (ERB3RDRS)” on page 137 for information about how to invoke the Data Retrieval Service.

The following ISPF shared variables contain information that RMF uses to generate a report during phase 1:

ERBREPC
The ISPF shared pool variable that contains the current command or report selection. RMF uses this variable as a key to ERBPHDT3, the phase driver table. This table has an entry (in the table field PHREPNA) for each RMF command or report selection. RMF retrieves the necessary information to generate the report during phase 1 from ERBPHDT3 (a copy of ERBPHDS3).

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

ERBSSHG3
The ISPF shared pool variable that contains the address of the set-of-samples header (SSHG3). This control block contains the addresses of the sample data that correspond to the time and range values specified during the Monitor III data gatherer session or as indicated by the Data Retrieval Service. (See Figure 24 on page 110 for an example of ERBSSHG3.)

ERBSUPP
The ISPF shared pool variable that contains the number of the subpool that non RMF functions must use for GETMAINS.

During phase 1, the phase driver module (ERB3RPH1) uses the information in the report row entry of ISPF table ERBPHDT3 (a copy of ERBPHDS3) to produce the RMF report. If you supplied your own program or CLIST for phase 1, that routine gets control.

Upon completion, phase 1 must provide the following output:

ERBREPC
The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET1
The ISPF shared pool variable that should contain your return code from the program or CLIST used in phase 1.

For RMF supplied reports, ERB3RPH1 creates the report in phase 1 and returns one of the following return codes:

0
ISPF table successfully generated for the report

4
ISPF table generated for the report has some data, but errors have occurred

8
ISPF table generated for the report has no data, and an error has occurred

For your own routine, you might want to use the same return codes.

PHDRTAB1
The ISPF shared pool variable that contains the name of the ISPF data table generated in phase 1. If you omit phase 2, you must ensure that PHDRTAB2 contains the same name as PHDRTAB1. See phases 2 and 3 described later.

You can define your own ISPF shared pool variables to contain information that you want to include for phase 1. To ensure that no interference with RMF created variables occurs, the first three characters of user-defined variables should appear as follows:

```
USR
```

Phase 2

For phase 2, you supply a program or CLIST to modify the ISPF table created for the report in phase 1.

The following ISPF shared variables contain information for phase 2:

ERBREPC
The ISPF shared pool variable that should contain the current command or report selection.

ERBCMDC,ERBPARC
The ISPF shared pool variables that should contain the current command (ERBCMDC) and any command parameters (ERBPARC).

PHDRET1
The ISPF shared pool variable that should contain your return code from the program or CLIST used in phase 1.

For RMF supplied reports, ERB3RPH1 creates the report in phase 1 and returns one of the following return codes:

0
ISPF table successfully generated for the report

4
ISPF table generated for the report has some data, but errors have occurred

8
ISPF table generated for the report has no data, and an error has occurred

For your own routine, you might want to use the same return codes.

PHDRTAB1
The ISPF shared pool variable that contains the name of the ISPF data table generated in phase 1.

Upon completion, phase 2 must provide the following output:

ERBREPC
The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET2
The ISPF shared pool variable that should contain the return codes from the RMF program or CLIST used to create the report in phase 2.

PHDRTAB2
The ISPF shared pool variable that should contain the name of the ISPF data table generated in phase 2. You can use the same table name as for PHDRTAB1.

Phase 3

For phase 3, RMF formats the ISPF table generated in phase 1 or 2 and displays the report. To format the ISPF report data tables, RMF uses the tabular report format table (ERBFMTS3), the RMF header table (ERBHDRS3), and the graphic parameter table (ERBPTGS3). The RMF display phase module (ERB3RDSP) displays the report by means of the ISPF TBDISPL service.

The following ISPF shared variables contain information that you can use to format and display a report during phase 3:

ERBREPC
The ISPF shared pool variable that contains the current command or report selection. The variable is a key to obtain formatting information for the tabular report in the report format table (ERBFMTS3) or the graphic report in the graphic parameter table (ERBPTGS3). For examples of these tables, see Appendix B.

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

PHDRET1,PHDRET2
The ISPF shared pool variables that should contain return codes from phase 1 and 2.

PHDRTAB2
The ISPF shared pool variable that should contain the name of the ISPF data table generated in phase 1 and/or phase 2.

SESRPFU3
The ISPF shared pool variable that contains the report mode (TABULAR or GRAPHIC).

RMF uses module ERB3RDSP to display the reports. The module dynamically constructs a panel from information in the format tables. It builds header and model lines and constructs the graphic area within the panel and uses the ISPF data table whose name appears in the ISPF shared pool variable PHDRTAB2.

Upon completion, phase 3 must provide the following output:

ERBREPC
The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET3
The ISPF shared pool variable that should contain the return code from the program or CLIST used to format and display the report.

If you decide to replace the RMF module ERB3RDSP with your own routine, you must consider the following:

- To obtain a display of your reports in GO mode, you must invoke the ISPF service CONTROL LOCK before the ISPF service TBDISPL is performed.
- Your module must handle all ISPF PASSTHRU commands.

Phase 4

For phase 4, you provide a program that can perform cleanup services for resources you might have used during previous phases. For example, if you have used ISPF TBCREATE with the WRITE SHARE option to create an ISPF table, you can use ISPF TBEND to delete the table during phase 4. Or use TBEND to save and then delete the table. See the ISPF publications that describe these services for more information.

The following ISPF shared variables contain information that you can use to format and display a report during phase 4:

ERBREPC
The ISPF shared pool variable that contains the current command or report selection.

ERBCMDC,ERBPARC
The ISPF shared pool variables that contain the current command (ERBCMDC) and any command parameters (ERBPARC).

ERBSUPP
The ISPF shared pool variable that contains the number of the subpool used for GETMAINS.

PHDRET1,PHDRET2,PHDRET3
The ISPF shared pool variables that should contain return codes from phase 1, 2, and 3.

Upon completion, phase 4 must provide the following output:

ERBREPC
The ISPF shared pool variable that should be restored to its value at entry to phase 1.

PHDRET4
The ISPF shared pool variables that should contain return codes from phase 4.

Data retrieval service (ERB3RDRS)

The Data Retrieval Service (ERB3RDRS) module provides flexibility for user exits to change the time range from which data is collected. The module is called from phase 1 of your user exit. This service can be invoked by either calling it,

Example 1

```
CALL ERB3RDRS (PARMAREA)
```

or by using the ISPF SELECT service.

Example 2

```
ISPEXEC SELECT PGM(ERB3RDRS) PARM(PARMAREA)
```

To use this service, the caller must invoke the module ERB3RDRS with the registers and parameter area described in “Parameter area contents” on page 138.

Programming considerations

Do not link the module ERB3RDRS to your application program. Assembler programs must use LOAD or LINK macros to access the module; PL/I programs must use FETCH/RELEASE; and C programs must use the built-in function FETCH.

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

Function codes

The function code specifies the time range to be used by the Data Retrieval Service:

1
Most recent number of MINTIMEs (as in GO mode)

2
Retrieve data from the range determined by BEG Date and Time and END Date and Time (similar to the BREF command with parameters DATE=,TIME=, and RANGE=)

3
Retrieve data from the range determined by using END Date and Time as end time, and going backward in time using the current RANGE (similar to the BREF command without parameters)

4
Retrieve data from the range determined by BEG Date and Time as begin time, and going forward in time using the current RANGE (similar to the FREF command without parameters)

Registers at entry

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

Register
Contents

0
Not used

1
Parameter list address

2-12
Not used

13
Standard save area address

14
Return address

15
Entry point address of ERB3RDRS

Parameter area contents

The parameter area passed by the caller to the RMF Data Retrieval Service is a single character string, preceded by a halfword containing the length of the parameter area in binary. The parameter area is as follows:

First word
Bytes 0 to 3: function code

Second word
Bytes 4 to 7: number of MINTIMEs (this is used only with function code 1)

Character string
Bytes 8 to 21: begin date and time of the requested time range in character format of MMDDYYYYHHMMSS.

Character string
Bytes 22 to 35: end date and time of the requested time range in character format of MMDDYYYYHHMMSS.

Output

The Data Retrieval Service module updates the following shared pool variables:

ERBSSHG3
The ISPF shared pool variable that contains the beginning address of the common set of samples. If no data could be retrieved, this variable is set to hexadecimal zero.

ERBDTBEG
The ISPF shared pool variable that contains the beginning date/time value of the retrieved range.

ERBDTEND
The ISPF shared pool variable that contains the ending date/time value of the retrieved range.

ERBMNTIM
The ISPF shared pool variable that contains the Monitor III data gatherer MINTIME option value in external format.

Return codes

Upon return from this service, register 15 provides the return code and reason code as listed in Table 5 on page 139:

- Bytes 0 and 1 are not used (x'0000')
- Byte 2 contains reason code
- Byte 3 contains return code

Table 5. Return and Reason Codes for the Data Retrieval Service (ERB3RDRS)

Return Code (Decimal)	Reason Code (Decimal)	Description
0	0	Data returned with no errors.
4	4	Data might be inconsistent due to a SET IPS change detected within the specified range. This is valid for data being gathered with RMF Version 4.
8		Data only partially returned.
	8	Partial data returned. Message ERB589I displayed.
	9	VSAM retrieval error occurred. Partial data returned. Message ERB589I displayed.
	13	The WLM service policy has changed, or the IPS values have been modified. This is valid for data being gathered with RMF Version 5 and above.
	14	The RMF cycle time has changed.
	15	IPL detected.

Table 5. Return and Reason Codes for the Data Retrieval Service (ERB3RDRS) (continued)		
Return Code (Decimal)	**Reason Code (Decimal)**	**Description**
12		No data returned.
	8	No data returned. Message ERB587I displayed.
	9	VSAM retrieval error occurred. No data returned. Message ERB587I displayed.
	14	Cycle time changed. Message ERB559I displayed.
	15	IPL detected. Message ERB558I displayed.
	16	No data available. Message ERB591I displayed
	17	Data gatherer is not active. Message ERB565I displayed.
	18	Preallocated data sets are unusable. Message ERB583I displayed.
	19	Preallocated data sets found to be unusable during data retrieval. Message ERB583I displayed.
	20	Too many reporters tried to access the in-storage buffer. Message ERB564I displayed.
	21	Retrieval from in-storage buffer failed. Message ERB564I displayed.
	22	No data is in the in-storage buffer. Message ERB591I displayed.
	23	Insufficient storage to copy data from the in-storage buffer. Message ERB564I displayed.
16	0	Incorrect function code.

Note: The RMF Monitor III standard reports provide information on the same time range as was requested in the last use of the Data Retrieval Service.

TSO/E user authorization

TSO/E must be installed on your system to use the following commands.

All the data collected and reported by RMF during a Monitor III display session is obtained from commonly addressable storage that is not fetch protected. However, if your installation wants to limit the use of the command that starts an RMF Monitor III session under TSO/E, one method available is to replace the RMF control section with your own module. For Monitor III you replace the control section ERB3SOCK. Your routine will then be invoked as part of the RMF response to the RMF command.

ERB3SOCK (Monitor III) runs in problem state with a key of 8. When this control section gets control, register 1 points to a two-word address list. The first address points to the eight-byte user ID of the user who has entered the RMF command. The second word points to the PSCB. Figure 46 on page 141 illustrates the input parameter structure.

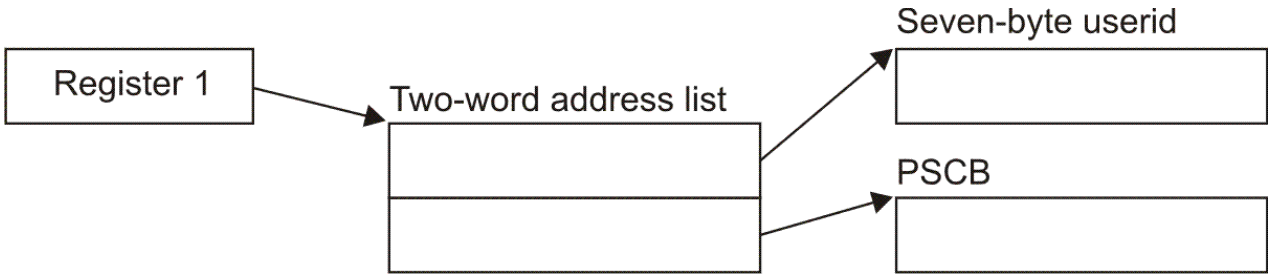


Figure 46. ERB3SOCK Input Parameter Structure

The module that you code to replace ERB3SOCK must be reenterable. It receives control by a BALR instruction and must save the registers when it receives control and restore the registers when it returns control. Register 13 contains the address of the register save area; register 14 contains the return address; and register 15 contains the entry address.

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

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

When your routine has completed its processing, set a return code of 0 in register 15 to indicate to RMF that the user is authorized to enter RMF. Set a return code of 4 in register 15 to indicate to RMF that the user is not authorized to enter RMF. In response to this return code, RMF will display a message at the display station. No session will be started. After setting the appropriate return code, return control by branching on the contents of register 14.

For the Monitor III TSO/E session the user authorization exit routine (ERB3SOCK) is part of the RMF load module that contains the RMF command. This module resides in SYS1.SERBLINK as load module RMF; its entry point is ERB3RTSO. Before your authorization routine can run, you must link edit it with RMF; the control statements required are:

```
(ERB3SOCK object deck)
INCLUDE ddname(RMF)
ENTRY ERB3RTSO
NAME RMF(R)
```

Chapter 5. Monitor III data reporter tables

This topic provides the following information:

- It describes the data tables, and graphic parameter table used by the Monitor III data reporter
- It lists the ISPF record fields and table entries associated with creating, formatting, and displaying RMF reports

See Chapter 4, “Adding Monitor III user exits,” on page 107 for information on how to create user-defined reports.

Tabular report format table ERBFMTS3

The RMF format table defines the layout of RMF reports for panel display and hardcopy printing. It also ensures that each output function within RMF produces the same format.

This table contains one row for each report name and format. Each row contains information on how to edit heading and column data and contains an example for each variable name.

Variable Name	T	Variable Description	Example
FMTREPNA	K	Report name	DELAY
FMTFORMT	K	Report format identifier (not yet used)	ENGLISH
FMTRMODE	N	Report mode available (GRAPHIC/TABULAR/ BOTH)	BOTH
FMTTPANL	N	Tabular report panel name	ERB3JDE
FMTTHLPP	N	Name of related help panel	ERB3JDE1
FMTLOGLN	N	Name of logical line number variable	JDEDTLLN
FMTSEQNR	N	Name of sequence number variable	JDEDTPSN
FMTCMDLN	N	Content of command line	COMMAND ===> &ZCMD ...
FMTHDR1	N	Content of header line 1 (text and variables intermixed)	... RMF DELAYS &HDRSID ..
FMTHDR2	N	Content of header line 2 (text and variables intermixed)	... Samples: &Z TIME: .
FMTSUBH1	N	Content of subheader line 1 (text and variables intermixed)	
FMTSUBH2	N	Content of subheader line 2 (text and variables intermixed)	
FMTSUBH3	N	Content of subheader line 3 (text and variables intermixed)	
FMTSUBH4	N	Content of subheader line 4 (text and variables intermixed)	
FMTSUBH5	N	Content of subheader line 5 (text and variables intermixed)	
FMTCOLH1	N	Text for column header line 1	WFL USG
FMTCOLH2	N	Text for column header line 2	NAME C DMN % % ..
FMTCOLH3	N	Text for column header line 3	
FMTHVPRE	N	Prefix used in specifying variables in header lines	&

Variable Name	T	Variable Description	Example
FMTHPLCH	N	Header line placeholder replacement variable names	HDRSAMPL HDRDATE HDRTIME
FMTSPLCH	N	Subheader line placeholder replacement variable names	
FMTCPLCH	N	Command line placeholder replacement variable names	AMT
FMTMODL1	N	Definition of model line 1 (attribute characters followed by variable names or placeholder values(Z), variable names used must be elements of the report column data table)	|JDELDAN|Z|Z |Z
FMTMODL2	N	Definition of model line 2	
FMTMODL3	N	Definition of model line 3	
FMTMATTR	N	Attribute characters used in model lines	_|¢
FMTMPLCH	N	Model line placeholder replacement variable names (ZVARS)	JDETYPE JDELDMN JDELPGN
FMTHVMAX	N	Number of variables within header lines (maximum of 20)	6
FMTSVMAX	N	Number of variables within subheader lines (maximum of 30)	0
FMTMVMAX	N	Number of variables within model lines (maximum of 30)	16
FMTCVMAX	N	Number of variables within command line (maximum of 5)	
FMTHVNnn	S	Variable name used in header lines	HDRSID
FMTHVRnn	S	Number of header line where variable is used	1
FMTHVPnn	S	Variable position within line	52
FMTHVLnn	S	Maximum variable length	15
FMTSVNxx	S	Variable name used in subheader lines	
FMTSVRxx	S	Number of subheader line where variable is used	
FMTSVPxx	S	Variable position within line	
FMTSVLxx	S	Maximum variable length	
FMTMVNyy	S	Variable name used in model lines	JDELDAN
FMTMVRyy	S	Number of model line where variable is used	1
FMTMVPyy	S	Variable position within line	2
FMTMVLyy	S	Maximum variable length	8
FMTCVNzz	S	Variable name used in command line	ZCMD
FMTCVPzz	S	Variable position within line	14
FMTCVLzz	S	Maximum variable length	51

Note:

K -
KEY type variable

N -
NAMES type variable

S -
EXTENSION type variable

nn =
unique number for each variable used in the header lines

xx =
unique number for each variable used in the subheader lines

yy =
unique number for each variable used in the model lines

zz =
unique number for each variable used in the command line

Header data table ERBHDRS3

The RMF header data table provides the variable heading information in one table row for each report.

Variable Name	T	Variable Description	Example
HDRREPNA	K	Report name	DELAY
ERBSID	N	System identifier	AQXA
ERBHCTXT	N	Hardcopy text constant	HARDCOPY
ERBSAMPL	N	Sample count	100
ERBDATE	N	Starting date	07/02/17
ERBTIME	N	Starting time	10.35.00
ERBRANGE	N	Time range value	100
ERBRMFVD	N	RMF version	RMF V2R4
ERBSPXID	N	Sysplex ID	RMFPLEX
ERBSNUM	N	Number of systems within sysplex	5
ERBSAMWL	N	Number of WLM samples	100
	S	The variable data for subheader lines has to be kept in extension values of this table. Example for STORR report.	

Note:

K -
KEY type variable

N -
NAMES type variable

S -
EXTENSION type variable

Monitor III data reporter tables

Each of the following report data tables indicates in column **Report** whether a value is part of the Monitor III report (Yes), is part of a pop-up window (Pop-Up), or is available through the Monitor III Utility (Util).

Column **Type** indicates whether it is a KEY-type variable (K) or a NAMES-type variable (N).

CACHDET - Tabular report data table ERBCADT3

RMF builds ERBCADT3 when using CACHDET as a report type.

Name	Type	Meaning	Report
CADDTLLN	K	Logical line number	-
CADDTPSN	K	Sequence number	-
CADPVOLU	N	Volume	Yes
CADPDEVN	N	Four-digit device number	Util
CADPDVN5	N	Five-digit device number	Yes
CADPSSID	N	SSID	Yes
CADPIOP	N	I/O percentage	Yes
CADPIO	N	I/O rate	Yes
CADPHITP	N	Hit percentage	Yes
CADPREAD	N	Cache hit rate READ	Yes
CADPDFW	N	Cache hit rate DFW	Yes
CADPCFW	N	Cache hit rate CFW	Yes
CADPTOT	N	DASD I/O rate total	Yes
CADPSTAG	N	DASD I/O rate stage	Yes
CADPSEQ	N	Sequential rate	Yes
CADPASYN	N	Async rate	Yes
CADVCACH	N	Cache state for volume	Pop-Up
CADVDFW	N	DFW state for volume	Pop-Up
CADVPIN	N	Pinned state for volume	Pop-Up
CADVNRRA	N	Norm Read rate for volume	Pop-Up
CADVNRHI	N	Norm Read hit rate for volume	Pop-Up
CADVNRHP	N	Norm Read hit percentage for volume	Pop-Up
CADVNWRA	N	Norm Write rate for volume	Pop-Up
CADVNWFA	N	Norm Write fast rate for volume	Pop-Up
CADVNWHI	N	Norm Write hit rate for volume	Pop-Up
CADVNWHP	N	Norm Write hit percentage for volume	Pop-Up
CADVNREP	N	Norm Read percentage for volume	Pop-Up
CADVNTRA	N	Norm Tracks rate for volume	Pop-Up
CADVSRRA	N	Seq Read rate for volume	Pop-Up
CADVSRHI	N	Seq Read hit rate for volume	Pop-Up
CADVSRHP	N	Seq Read hit percentage for volume	Pop-Up
CADVSWRA	N	Seq Write rate for volume	Pop-Up
CADVSWFA	N	Seq Write fast rate for volume	Pop-Up
CADVSWHI	N	Seq Write hit rate for volume	Pop-Up
CADVSWHP	N	Seq Write hit percentage for volume	Pop-Up
CADVSREP	N	Seq Read percentage for volume	Pop-Up
CADVSTRA	N	Seq Tracks rate for volume	Pop-Up
CADVCRRA	N	CFW Read rate for volume	Pop-Up
CADVCRHI	N	CFW Read hit rate for volume	Pop-Up
CADVCRHP	N	CFW Read hit percentage for volume	Pop-Up

Name	Type	Meaning	Report
CADVCWRA	N	CFW Write rate for volume	Pop-Up
CADVCWFA	N	CFW Write fast rate for volume	Pop-Up
CADVCWHI	N	CFW Write hit rate for volume	Pop-Up
CADVCWHP	N	CFW Write hit percentage for volume	Pop-Up
CADVCREP	N	CFW Read percentage for volume	Pop-Up
CADVTRRA	N	Total Read rate for volume	Pop-Up
CADVTRHI	N	Total Read hit rate for volume	Pop-Up
CADVTRHP	N	Total Read hit percentage for volume	Pop-Up
CADVTWRA	N	Total Write rate for volume	Pop-Up
CADVTWFA	N	Total Write fast rate for volume	Pop-Up
CADVTWHI	N	Total Write hit rate for volume	Pop-Up
CADVTWHP	N	Total Write hit percentage for volume	Pop-Up
CADVTREP	N	Total Read percentage for volume	Pop-Up
CADVMMDB	N	Delayed Operations due to NVS for volume	Pop-Up
CADVMNCI	N	Non-cache ICL for volume	Util
CADVMCWR	N	CKD write for volume	Pop-Up
CADVMCRM	N	Read miss for volume	Pop-Up
CADVMMCB	N	Delayed Operations due to Cache for volume	Pop-Up
CADVMNCB	N	Non-cache bypass for volume	Util
CADVMCHI	N	CKD hits for volume	Pop-Up
CADVMCWP	N	Write prom for volume	Pop-Up
CADVMMDI	N	DFW inhibit for volume	Pop-Up
CADSNRRA	N	Norm Read rate for SSID	Pop-Up
CADSNRHI	N	Norm Read hit rate for SSID	Pop-Up
CADSNRHP	N	Norm Read hit percentage for SSID	Pop-Up
CADSNWRA	N	Norm Write rate for SSID	Pop-Up
CADSNWFA	N	Norm Write fast rate for SSID	Pop-Up
CADSNWHI	N	Norm Write hit rate for SSID	Pop-Up
CADSNWHP	N	Norm Write hit percentage for SSID	Pop-Up
CADSNREP	N	Norm Read percentage for SSID	Pop-Up
CADSNTRA	N	Norm Tracks rate for SSID	Pop-Up
CADSSRRA	N	Seq Read rate for SSID	Pop-Up
CADSSRHI	N	Seq Read hit rate for SSID	Pop-Up
CADSSRHP	N	Seq Read hit percentage for SSID	Pop-Up
CADSSWRA	N	Seq Write rate for SSID	Pop-Up
CADSSWFA	N	Seq Write fast rate for SSID	Pop-Up
CADSSWHI	N	Seq Write hit rate for SSID	Pop-Up
CADSSWHP	N	Seq Write hit percentage for SSID	Pop-Up
CADSSREP	N	Seq Read percentage for SSID	Pop-Up
CADSSTRA	N	Seq Tracks rate for SSID	Pop-Up

Name	Type	Meaning	Report
CADSCRRA	N	CFW Read rate for SSID	Pop-Up
CADSCRHI	N	CFW Read hit rate for SSID	Pop-Up
CADSCRHP	N	CFW Read hit percentage for SSID	Pop-Up
CADSCWRA	N	CFW Write rate for SSID	Pop-Up
CADSCWFA	N	CFW Write fast rate for SSID	Pop-Up
CADSCWHI	N	CFW Write hit rate for SSID	Pop-Up
CADSCWHP	N	CFW Write hit percentage for SSID	Pop-Up
CADSCREP	N	CFW Read percentage for SSID	Pop-Up
CADSTRRA	N	Total Read rate for SSID	Pop-Up
CADSTRHI	N	Total Read hit rate for SSID	Pop-Up
CADSTRHP	N	Total Read hit percentage for SSID	Pop-Up
CADSTWRA	N	Total Write rate for SSID	Pop-Up
CADSTWFA	N	Total Write fast rate for SSID	Pop-Up
CADSTWHI	N	Total Write hit rate for SSID	Pop-Up
CADSTWHP	N	Total Write hit percentage for SSID	Pop-Up
CADSTREP	N	Total Read percentage for SSID	Pop-Up
CADSMMDB	N	Delayed Operations due to NVS for SSID	Pop-Up
CADSMNCI	N	Non-cache ICL for SSID	Util
CADSMCWR	N	CKD write for SSID	Pop-Up
CADSMCRM	N	Read miss for SSID	Pop-Up
CADSMMCB	N	Delayed Operations due to Cache for SSID	Pop-Up
CADSMNCB	N	Non-cache bypass for SSID	Util
CADSMCHI	N	CKD hits for SSID	Pop-Up
CADSMCWP	N	Write prom for SSID	Pop-Up
CADSMMDI	N	DFW inhibit for SSID	Pop-Up

CACHSUM - Tabular report data table ERBCAST3

RMF builds ERBCAST3 when using CACHSUM as a report type.

Name	Type	Meaning	Report
CASDTLLN	K	Logical line number	-
CASDTPSN	K	Sequence number	-
CASPSSID	N	SSID	Yes
CASPCUID	N	CUID	Yes
CASPTYPM	N	Type-Mod	Yes
CASPSIZE	N	Storage size	Yes
CASPIO	N	I/O rate	Yes
CASPHITP	N	Hit percentage	Yes
CASPHIT	N	Hit rate	Yes
CASPMTOT	N	Miss total rate	Yes
CASPMSTG	N	Miss stage rate	Yes

Name	Type	Meaning	Report
CASPREAP	N	Read percentage	Yes
CASPSEQ	N	Sequential rate	Yes
CASPASYN	N	Async rate	Yes
CASPOFF	N	Off rate	Util
CASNRRA	N	Norm Read rate	Pop-Up
CASNRHI	N	Norm Read hit rate	Pop-Up
CASNRHIP	N	Norm Read hit percentage	Pop-Up
CASNWRA	N	Norm Write rate	Pop-Up
CASNWFA	N	Norm Write fast rate	Pop-Up
CASNWHI	N	Norm Write hit rate	Pop-Up
CASNWHIP	N	Norm Write hit percentage	Pop-Up
CASNREAP	N	Norm Read percentage	Pop-Up
CASNTRA	N	Norm Tracks rate	Pop-Up
CASSRRA	N	Seq Read rate	Pop-Up
CASSRHI	N	Seq Read hit rate	Pop-Up
CASSRHIP	N	Seq Read hit percentage	Pop-Up
CASSWRA	N	Seq Write rate	Pop-Up
CASSWFA	N	Seq Write fast rate	Pop-Up
CASSWHI	N	Seq Write hit rate	Pop-Up
CASSWHIP	N	Seq Write hit percentage	Pop-Up
CASSREAP	N	Seq Read percentage	Pop-Up
CASSTRA	N	Seq Tracks rate	Pop-Up
CASCRRA	N	CFW Read rate	Pop-Up
CASCRHI	N	CFW Read hit rate	Pop-Up
CASCRHIP	N	CFW Read hit percentage	Pop-Up
CASCWRA	N	CFW Write rate	Pop-Up
CASCWFA	N	CFW Write fast rate	Pop-Up
CASCWHI	N	CFW Write hit rate	Pop-Up
CASCWHIP	N	CFW Write hit percentage	Pop-Up
CASCREAP	N	CFW Read percentage	Pop-Up
CASTRRA	N	Total Read rate	Pop-Up
CASTRHI	N	Total Read hit rate	Pop-Up
CASTRHIP	N	Total Read hit percentage	Pop-Up
CASTWRA	N	Total Write rate	Pop-Up
CASTWFA	N	Total Write fast rate	Pop-Up
CASTWHI	N	Total Write hit rate	Pop-Up
CASTWHIP	N	Total Write hit percentage	Pop-Up
CASTREAP	N	Total Read percentage	Pop-Up
CASMCACH	N	Cache state	Util
CASMCCON	N	Cache configured	Pop-Up

Name	Type	Meaning	Report
CASMCAVL	N	Cache available	Pop-Up
CASMCOFF	N	Cache offline	Pop-Up
CASMCPIN	N	Cache pinned	Pop-Up
CASMNVS	N	NVS state	Pop-Up
CASMNCON	N	NVS configured	Pop-Up
CASMNPIN	N	NVS pinned	Pop-Up

CFACT - Tabular report data table ERBCFAT3

RMF builds ERBCFAT3 when using CFACT as a report type.

Name	Type	Meaning	Report
CFADTLLN	K	Logical line number	-
CFADTPSN	K	Sequence number	-
CFAPSTRU	N	Structure name	Yes
CFAPTYPE	N	Structure type	Yes
CFAPSTAT	N	Structure status	Yes
CFAPSTEX	N	Structure status extension	Util
CFAPENCR	N	Encryption indicator of CF structure	Yes
CFAPSYS	N	System name	Yes
CFAPSTEP	N	Structure execution %	Util
CFAPUTIP	N	CPU utilization %	Yes
CFAPSYNR	N	Sync rate	Yes
CFAPASS	N	Sync average service time	Yes
CFAPSYNC	N	Number of synchronous requests	Util
CFAPASYR	N	Async rate	Yes
CFAPAAS	N	Async average service time	Yes
CFAPASYC	N	Number of asynchronous requests	Util
CFAPACHG	N	Async changed %	Yes
CFAPADEL	N	Async delay %	Yes
CFAPQRT	N	Average queued request time	Util
CFAPCNVC	N	Converted request count	Util
CFAPDELC	N	Operation count delayed for dump serialization	Util
CFAPQUEC	N	Queued operation count	Util
CFAPMUSR	N	Maximum number of users	Util
CFAPTUSR	N	Total number of users	Util
CFAPPUSR	N	Number of problem users	Util
CFAPREBP	N	Rebuild percentage	Util
CFAINAM	N	Coupling facility name	Yes
CFAISTRU	N	Structure name	Pop-Up
CFAITYPE	N	Structure type	Pop-Up
CFAICNAM	N	Connection name	Pop-Up

Name	Type	Meaning	Report
CFAICJOB	N	Job name	Pop-Up
CFAICSTA	N	Status	Pop-Up
CFAICPRB	N	Problem status	Util
CFAICASI	N	ASID	Pop-Up
CFAICLVL	N	CF level	Pop-Up
CFAICREB	N	User managed rebuild allowed	Util
CFAICDRB	N	User managed rebuild with duplexing allowed	Util
CFAICALT	N	Altering allowed	Util
CFAICAUT	N	System managed processes allowed	Util
CFAICSUS	N	Suspension of work is tolerated	Util
CFAISTRS	N	Structure size	Pop-Up
CFAISTRP	N	Structure size %	Util
CFAISTUP	N	Structure utilized storage %	Util
CFAISTRC	N	Structure storage class	Util
CFAISTRM	N	Min structure size	Util
CFAISTRX	N	Max structure size	Util
CFAIDTS	N	Dump table size	Util
CFAILDES	N	Data element size (LIST/LOCK only)	Util
CFAILDLS	N	Data list entry size (LIST/LOCK only)	Util
CFAILEL	N	List entries total (LIST/LOCK only)	Pop-Up
CFAILEM	N	List entries current (LIST/LOCK only)	Pop-Up
CFAIMAE	N	Data elements total (LIST only)	Pop-Up
CFAICUE	N	Data elements current (LIST only)	Pop-Up
CFAILTL	N	Lock entries total (LIST/LOCK only)	Pop-Up
CFAILTM	N	Lock entries current (LIST/LOCK only)	Pop-Up
CFAIDES	N	Data element size (CACHE only)	Util
CFAIDEN	N	Directory entries total (CACHE only)	Pop-Up
CFAIDEC	N	Directory entries current (CACHE only)	Pop-Up
CFAICEN	N	Directory entries changed (CACHE only)	Util
CFAIDEL	N	Data elements total (CACHE only)	Pop-Up
CFAIDAC	N	Data elements current (CACHE only)	Pop-Up
CFAICEL	N	Data elements changed (CACHE only)	Util
CFAICONT	N	Contention %	Pop-Up
CFAIFCON	N	False Contention % (LOCK only)	Pop-Up
CFAIREQR	N	Request rate (CACHE only)	Pop-Up
CFAIREAR	N	Read rate (CACHE only)	Pop-Up
CFAIWRIR	N	Write rate (CACHE only)	Pop-Up
CFAICAOR	N	Castout rate (CACHE only)	Pop-Up
CFAIXIR	N	XI rate (CACHE only)	Pop-Up
CFAIDER	N	Directory reclaims (CACHE only)	Pop-Up

Name	Type	Meaning	Report
CFAIFCCL	N	First castout class	Util
CFAILCCL	N	Last castout class	Util
CFAIPREF	N	Allocation preference list	Util
CFAIEXCL	N	Exclusion preference list	Util
CFAISPCF	N	% of CF storage	Pop-Up
CFAISAUM	N	Estimated Max. of Augmented Space (LIST only)	Pop-Up
CFAISAUP	N	% Augmented Space Used (LIST only)	Pop-Up
CFAISSCM	N	SCM Space Maximum (LIST only)	Pop-Up
CFAISSCP	N	SCM Space % Used (LIST only)	Pop-Up
CFAISLTM	N	SCM List Entries Est Max (LIST only)	Pop-Up
CFAISLTC	N	SCM List Entries Current (LIST only)	Pop-Up
CFAISLMM	N	SCM List Elements Est Max (LIST only)	Pop-Up
CFAISLMC	N	SCM List Elements Current (LIST only)	Pop-Up
CFAISALG	N	SCM Algorithm Type	Pop-Up

CFOVER - Tabular report data table ERBCFOT3

RMF builds ERBCFOT3 when using CFOVER as a report type.

Name	Type	Meaning	Report
CFODTLLN	K	Logical line number	-
CFODTPSN	K	Sequence number	-
CFOPNAM	N	Coupling facility name	Yes
CFOPMOD	N	Model	Yes
CFOPVER	N	Version	Yes
CFOPLVL	N	CF level	Yes
CFOPDYND	N	CF dynamic dispatching	Yes
CFOPSTAT	N	Status of CF	Util
CFOPVOL	N	Volatility status	Util
CFOPUTIP	N	Processor utilization %	Yes
CFOPDEF	N	Processor defined	Yes
CFOPPDED	N	Number of dedicated processors	Util
CFOPPSHR	N	Number of shared processors	Yes
CFOPPWGT	N	Average weighting of shared processors	Yes
CFOPEFF	N	Processor effective	Yes
CFOPREQR	N	Request rate	Yes
CFOPTSD	N	Storage size	Yes
CFOPTSF	N	Storage available	Yes
CFOPUTIS	N	Utilized storage %	Util
CFOPTCS	N	Total control space	Util
CFOPFCS	N	Free control space	Util
CFOPDTS	N	Dump table control space	Util

Name	Type	Meaning	Report
CFOPDTUS	N	Dump table in use	Util
CFOPSYSC	N	Connected MVS system count	Util
CFOPSTCI	N	Structure count in policy	Util
CFOPSTCO	N	Structure count out policy	Util
CFOPMNT	N	Maintenance mode active	Util
CFOPRCV	N	Recovery manager active	Util
CFOPSCMS	N	Storage Class Memory size	Yes
CFOPSCMA	N	Storage Class Memory available	Yes
CFOPSCMU	N	Utilized Storage Class Memory %	Util
CFOPAUGS	N	Augmented space maximum	Util
CFOPAUGA	N	Augmented space available	Util
CFOPAUGU	N	% Utilized augmented space	Util
CFOPSMSC	N	Sum of maximum storage class memory	Util

CFSYS - Tabular report data table ERBCFST3

RMF builds ERBCFST3 when using CFSYS as a report type.

Name	Type	Meaning	Report
CFSDTLLN	K	Logical line number	-
CFSDTPSN	K	Sequence number	-
CFSPNAM	N	Coupling facility name	Yes
CFSPSYS	N	System name	Yes
CFSPSDEL	N	Subchannel delay %	Yes
CFSPSBSP	N	Subchannel busy %	Yes
CFSPPTHA	N	Paths available	Yes
CFSPPDEL	N	Paths delay %	Yes
CFSPSYNR	N	Sync rate	Yes
CFSPASS	N	Sync average service time	Yes
CFSPSYNC	N	Synchronous request count	Util
CFSPSOPD	N	Average synchronous operation delay	Util
CFSPSYNP	N	Synchronous request %	Util
CFSPASYR	N	Async rate	Yes
CFSPAAS	N	Async average service time	Yes
CFSPASYC	N	Asynchronous request count	Util
CFSPACHG	N	Async changed %	Yes
CFSPADEL	N	Async delay %	Yes
CFSPASYP	N	Asynchronous request %	Util
CFSPREQC	N	Total request %	Util
CFSPFOPT	N	Average failed operation time	Util
CFSPCNVC	N	Synchronous to asynchronous conversion rate	Util
CFSINAM	N	Coupling facility name	Pop-Up

Name	Type	Meaning	Report
CFSISCG	N	Subchannels generated	Pop-Up
CFSISCU	N	Subchannels in use	Pop-Up
CFSISCL	N	Subchannels max	Pop-Up
CFSIPATH	N	Paths IDs	Util
CFSIPTYP	N	TYPES variable string	Util
CFSITYPE	N	Path types	Util
CFSICPIn	N	Channel path ID (n^{th} of eight)	Pop-Up
CFSICPTn	N	Channel path type (n^{th} of eight)	Pop-Up
CFSICPOn	N	Channel path operation mode (n^{th} of eight)	Pop-Up
CFSICPDn	N	Degraded Mode indicator for the channel path (n^{th} of eight)	Pop-Up
CFSICPLn	N	Estimated distance in kilometers (n^{th} of eight)	Pop-Up
CFSIPHYn	N	Physical channel path ID (n^{th} of eight)	Pop-Up
CFSIHCAn	N	Host communication adapter ID (n^{th} of eight)	Pop-Up
CFSIHCPn	N	Host communication adapter port number (n^{th} of eight)	Pop-Up
CFSIIOPn	N	IOP ID (n^{th} of eight)	Pop-Up

CHANNEL - Tabular report data table ERBCHAT3

RMF builds ERBCHAT3 when using CHANNEL as a report type.

Name	Type	Meaning	Report
CHADTLLN	K	Logical line number	-
CHADTPSN	K	Sequence number	-
CHACPIVC	N	Channel path ID	Yes
CHACPNVC	N	Number of DCM-managed channels	Yes
CHACGVC	N	Channel type generation	Yes
CHACPTVC	N	Channel path type	Yes
CHACSIVC	N	Channel shared indication	Yes
CHACPUVC	N	Partition utilization percent	Yes
CHACTUVC	N	Total utilization percent	Yes
CHACTBVC	N	Bus utilization percent	Yes
CHACPRVC	N	Partition transfer rate (Read) in B/sec	Yes
CHACTRVC	N	Total transfer rate (Read) in B/sec	Yes
CHACPWVC	N	Partition transfer rate (Write) in B/sec	Yes
CHACTWVC	N	Total transfer rate (Write) in B/sec	Yes
CHACPMVC	N	Partition message sent rate	Util
CHACTMVC	N	Total message sent rate	Util
CHACPSVC	N	Partition message sent size	Util
CHACTSVC	N	Total message sent size	Util
CHACSFVC	N	Partition message sent fail rate	Util
CHACPFVC	N	Partition message receive fail rate	Util
CHACTFVC	N	Total message receive fail rate	Util

Name	Type	Meaning	Report
CHACFRTE	N	Rate of native FICON operations	Yes
CHACFACT	N	Average number of native FICON operations concurrently active	Yes
CHACXRTE	N	Rate of High Performance FICON (zHPF) operations	Yes
CHACXACT	N	Average number of zHPF operations concurrently active	Yes
CHACFDFR	N	Number of deferred native FICON operations per second	Util
CHACXDFR	N	Number of deferred zHPF operations per second	Util
CHACNET1	N	Physical-network identifier (PNET ID) of first channel path port	Util
CHACNET2	N	Physical-network identifier (PNET ID) of second channel path port	Util

CPC - Tabular report data table ERBCPCT3

RMF builds ERBCPCT3 when using CPC as a report type.

Name	Type	Meaning	Report
CPCDTLLN	K	Logical line number	-
CPCDTPSN	K	Sequence number	-
CPCPBIIP	N	zIIP boost active at end of MINTIME	Util
CPCPBSPD	N	Speed boost active at end of MINTIME	Util
CPCPPNAM	N	Partition name	Yes
CPCPDMSU	N	Defined capacity limit	Yes
CPCPAMSU	N	Actual consumed MSUs	Yes
CPCPCAPD	N	Hardware capping options of this partition (Y=yes, N=no)	Yes
CPCPCAPI	N	Initial capping (yes/no/mix)	Util
CPCPHWCC	N	Absolute physical hardware capacity limit in numbers of CPUs	Util
CPCPLPNO	N	Average number of logical processors or cores	Yes
CPCPLEFU	N	Logical processor effective utilization %	Yes
CPCPLTOU	N	Logical processor total utilization %	Yes
CPCPPLMU	N	Physical LPAR utilization %	Yes
CPCPPEFU	N	Physical processor effective utilization %	Yes
CPCPPTOU	N	Physical processor total utilization %	Yes
CPCPIND	N	Type/partition indicator	No
CPCPLPND	N	Number of logical processors or cores defined	Util
CPCPWGHT	N	Current weighting of shared CPU resources	Util
CPCPDEDP	N	Number of dedicated processors online	Util
CPCPLPSH	N	Percentage of the physical processor that a logical processor of the LPAR is entitled to use. If HiperDispatch is active, this is the percentage of logical processors with medium share.	Util
CPCPVCMH	N	If HiperDispatch is active, this is the number of logical processors or cores with high share.	Util
CPCPVCMM	N	If HiperDispatch is active, this is the number of logical processors or cores with medium share.	Util
CPCPVCML	N	If HiperDispatch is active, this is the number of logical processors or cores with low share.	Util
CPCPOSNM	N	Operating system name	Util
CPCPLPCN	N	LPAR cluster name	Util

Name	Type	Meaning	Report
CPCPLCIW	N	Initial weight defined	Util
CPCPLCMW	N	Minimum weight defined	Util
CPCPLCXW	N	Maximum weight defined	Util
CPCPCGNM	N	Group capacity name	Util
CPCPCGLT	N	Group capacity limit	Util
CPCPCGEM	N	Group minimum entitlement	Util
CPCPCGEX	N	Group maximum entitlement	Util
CPCPCSMB	N	Central storage in MB	Util
CPCPUPID	N	User partition ID	Util
CPCPHGNM	N	Hardware group name to which this partition belongs	Util
CPCPHWGC	N	Absolute hardware group capping limit for members of the same hardware group in numbers of CPUs	Util

Fields in the CPC report header

Name	Description of the variable	Report
CPCHPNAM	Name of partition that collected the data	Yes
CPCHMOD	Processor type	Yes
CPCHMDL	Processor model	Yes
CPCHCMSU	Effective CPC capacity (MSU/hour)	Yes
CPCHCCAI	Capacity adjustment indication	Util
CPCHCCCR	Capacity change reason	Util
CPCHWF	Weight % of Max	Yes
CPCHLMSU	4h MSU average	Yes
CPCHGNAM	Capacity group name	Yes
CPCHIMSU	Image capacity	Yes
CPCHCAP	WLM capping %	Yes
CPCHLMAX	4h MSU maximum	Yes
CPCHGLIM	Capacity group limit	Yes
CPCHGL4H	< 4h indicator for group	Yes
CPCHAMSU	Absolute MSU capping is active (Y / N)	Yes
CPCHRMSU	Time until capping	Util
CPCHRGRP	Time until capacity group is subject to capping	Util
CPCHGAUN	Capacity group average unused service units	Util
CPCHCPU	CPC sequence number	Util
CPCHCPCN	CPC name	Util
CPCHCPNO	Number of physical CPs	Util
CPCHIFAN	Number of physical zAAPs	Util
CPCHICFN	Number of physical ICFs	Util
CPCHIFLN	Number of physical IFLs	Util
CPCHSUPN	Number of physical ZIIPs	Util
CPCHPANO	Number of configured LPARs	Util

Name	Description of the variable	Report
CPCHWAIT	Wait completion indicator	Util
CPCHPMSU	% capacity used	Util
CPCHDEDC	Number of dedicated CPs across CPC	Util
CPCHDEDA	Number of dedicated zAAPs across CPC	Util
CPCHDEDI	Number of dedicated ZIIPs across CPC	Util
CPCHSHRC	Number of shared physical CPs across CPC	Util
CPCHSHRA	Number of shared physical zAAPs across CPC	Util
CPCHSHRI	Number of shared physical ZIIPs across CPC	Util
CPCHCUTL	% total physical utilization of shared CPs	Util
CPCHAUTL	% total physical utilization of shared zAAPs	Util
CPCHUUTL	% total physical utilization of shared ZIIPs	Util
CPCHLUTL	% total physical utilization of shared ICFs	Util
CPCHFUTL	% total physical utilization of shared IFLs	Util
CPCHVCPU	VARYCPU option (YES/NO)	Util
CPCHWMGT	WLM management (YES/NO)	Util
CPCHPRDS	Multithreading IIP core productivity	Yes
CPCHPRD	Multithreading core productivity for general purpose processors	Util
CPCHMCFS	Multithreading Maximum Capacity Factor for IIP	Util
CPCHMCF	Multithreading Maximum Capacity Factor for general purpose processors	Util
CPCHMTMS	Multithreading Mode for IIP	Yes
CPCHMTM	Multithreading Mode for general purpose processors	Util
CPCHCFS	Multithreading Capacity Factor for IIP	Util
CPCHCF	Multithreading Capacity Factor for general purpose processors	Util
CPCHATDS	Average Thread Density for IIP	Util
CPCHATD	Average Thread Density for general purpose processors	Util
CPCHBSTT	Boost type active at the end of MINTIME	Yes
CPCHBSTC	Boost class active at the end of MINTIME	Yes

CRYPTO - Tabular report data table ERBCRYT3

RMF builds ERBCRYT3 when using CRYOVW, CRYACC, or CRYPKC as a report type.

Name	T	Description of the variable	Report
CRYDTLLN	K	Logical line number	-
CRYDTPSN	K	Sequence number	-
CRYCTYPE	N	Type of cryptographic hardware function	Yes
CRYCIDX	N	Index of cryptographic hardware function	Yes
CRYCPC	N	CPC name	Yes
CRYSYS	N	System name	Yes
CRYUDID	N	Usage domain ID	Util
CRYSCPM	N	Scope of performance metrics (C, S)	Util
CRYCMODE	N	Cryptographic mode	Util

Name	T	Description of the variable	Report
CRYOPRT	N	Rate of all operations	Yes
CRYOPET	N	Average execution time (in milliseconds) of all operations	Yes
CRYTUTL	N	Total utilization percentage	Yes
CRCKGORT	N	Rate for RSA-key-generation operations (CCA coprocessor mode)	Yes
CRCKGOET	N	Average execution time (in milliseconds) for RSA-key-generation operations (CCA coprocessor mode)	Yes
CRCKGUTL	N	Utilization percentage for RSA-key-generation operations (CCA coprocessor mode)	Yes
CRARKLEN	N	RSA key length (accelerator mode)	Yes
CRAMOPRT	N	Rate of ME-format RSA operations (accelerator mode)	Yes
CRAMOPET	N	Average execution time (in milliseconds) of ME-format RSA operations (accelerator mode)	Yes
CRAMUTL	N	Utilization percentage of ME-format RSA operations (accelerator mode)	Yes
CRACOPRT	N	Rate of CRT-format RSA operations (accelerator mode)	Yes
CRACOPET	N	Average execution time (in milliseconds) of CRT-format RSA operations (accelerator mode)	Yes
CRACUTL	N	Utilization percentage of CRT-format RSA operations (accelerator mode)	Yes
CRPSART	N	Rate of slow asymmetric-key operations (PKCS11 coprocessor mode)	Yes
CRPSAET	N	Average execution time (in milliseconds) of slow asymmetric-key operations (PKCS11 coprocessor mode)	Util
CRPSAUTL	N	Utilization percentage of slow asymmetric-key operations (PKCS11 coprocessor mode)	Yes
CRPFART	N	Rate of fast asymmetric-key operations (PKCS11 coprocessor mode)	Yes
CRPFAET	N	Average execution time (in milliseconds) of fast asymmetric-key operations (PKCS11 coprocessor mode)	Util
CRPFAUTL	N	Utilization percentage of fast asymmetric-key operations (PKCS11 coprocessor mode)	Yes
CRPSPRT	N	Rate of symmetric-key operations that return partial or incremental results (PKCS11 coprocessor mode)	Yes
CRPSPET	N	Average execution time (in milliseconds) of symmetric-key operations that return partial or incremental results (PKCS11 coprocessor mode)	Util
CRSPUTL	N	Utilization percentage of symmetric-key operations that return partial or incremental results (PKCS11 coprocessor mode)	Yes
CRPSCRT	N	Rate of symmetric-key operations that return complete or final results (pkcs11 coprocessor mode)	Yes
CRPSCET	N	Average execution time (in milliseconds) of symmetric-key operations that return complete or final results (PKCS11 coprocessor mode)	Util
CRPSCUTL	N	Utilization percentage of symmetric-key operations that return complete or final results (PKSC11 coprocessor mode)	Yes
CRPAGRT	N	Rate of asymmetric-key generation operations (PKCS11 coprocessor mode)	Util
CRPAGET	N	Average execution time (in milliseconds) of asymmetric-key generation operations (PKCS11 coprocessor mode)	Util
CRPAGUTL	N	Utilization percentage of asymmetric-key generation operations (PKCS11 coprocessor mode)	Util
CRYSID	N	SMF system ID	Util

DELAY - Tabular report data table ERBJDET3

RMF builds ERBJDET3 when using DELAY as a report type.

Name	Type	Meaning	Report
JDEDTLLN	K	Logical line number	-
JDEDTPSN	K	Sequence number	-
JDELDAN	N	Jobname or summary	Yes
JDETYPE	N	Class (A, B, E, O, S, or T)	Util
JDETYPX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
JDELDMN	N	Domain number; no longer used	Yes
JDELPGN	N	Performance group number; no longer used	Yes
JDEPSVCL	N	Service class name	Yes
JDEGMIP	N	Indicator whether Storage Critical and/or CPU Critical (S, C, or SC)	Yes
JDELWFL	N	Work flow percentage	Yes
JDELUSG	N	Using percentage	Yes
JDELDEL	N	Delay percentage	Yes
JDELIDL	N	Idle percentage	Yes
JDELUKN	N	Unknown percentage	Yes
JDELPROC	N	Processor delay percentage	Yes
JDELDEV	N	Device delay percentage	Yes
JDELSTOR	N	Storage delay percentage	Yes
JDELSUBS	N	JES, HSM, and XCF delay percentage	Yes
JDELOPER	N	Operator delay percentage	Yes
JDELENQ	N	ENQ delay percentage	Yes
JDELJES	N	JES delay percentage	Util
JDELHSM	N	HSM delay percentage	Util
JDELXCF	N	XCF delay percentage	Util
JDELMNT	N	Operator mount delay percentage	Util
JDELMES	N	Operator message delay percentage	Util
JDELQUI	N	Operator quiesce delay percentage	Util
JDELREAS	N	Primary reason	Yes
JDELCAP	N	CAP delay %	Util
JDELCP	N	CP delay %	Util
JDELCBP	N	CBP delay %	Util
JDELSUP	N	zIIP delay %	Util
JDELIFA	N	zAAP delay %	Util

DEV - Tabular report data table ERBDEVT3

RMF builds table ERBDEVT3 when using DEV as a report type.

Name	Type	Meaning	Report
DEVDTLLN	K	Logical line number	-
DEVDTPSN	K	Sequence number	-
DEVPJOB	N	Jobname	Yes
DEVPCLA	N	Class (A, B, O, S, or T)	Yes

Name	Type	Meaning	Report
DEVPDMN	N	Domain number; no longer used	Yes
DEVPPGN	N	Performance group number; no longer used	Yes
DEVPSVCL	N	Service class name	Yes
DEVPODEL	N	Overall delay percentage	Yes
DEVPOUSE	N	Overall using percentage	Yes
DEVPCON	N	Connect time	Yes
DEV1SDEL	N	Delay percentage causes by volser1	Yes
DEV1VOLU	N	Volume serial number volser1	Yes
DEV2SDEL	N	Delay percentage caused by volser2	Yes
DEV2VOLU	N	Volume serial number volser2	Yes
DEV3SDEL	N	Delay percentage cause by volser3	Yes
DEV3VOLU	N	Volume serial number volser3	Yes
DEV4SDEL	N	Delay percentage caused by volser4	Yes
DEV4VOLU	N	Volume serial number volser4	Yes

DEVR - Tabular report data table ERBDVRT3

RMF builds ERBDVRT3 when using DEVR as a report type.

Name	Type	Meaning	Report
DVRDTLLN	K	Logical line number	-
DVRDTPSN	K	Sequence number	-
DVRPVOLU	N	Volser	Yes
DVRPDEVN	N	Four-digit device number	Util
DVRPDVN5	N	Five-digit device number	Yes
DVRPIDEN	N	Device indication (model)	Yes
DVRPSTAT	N	Status	Yes
DVRPEXP	N	Number of exposures	Yes
DVRPACTV	N	Percentage of active time	Yes
DVRPCONN	N	Percentage of connect time	Yes
DVRPDISC	N	Percentage of disconnect time	Yes
DVRPPEND	N	Percentage of pending time	Util
DVRPDLYR	N	Pending delay reason header	Yes
DVRPDLYP	N	Pending delay reason percentage	Yes
DVRACTRT	N	Device activity rate	Yes
DVRRESPT	N	Response Time	Yes
DVRIOSQT	N	IOS queue time	Util
DVRPDVBT	N	Percentage of device busy delay time	Util
DVRPCUBT	N	Percentage of control unit busy delay time	Util
DVRPSPBT	N	Percentage of director port busy delay time	Util
DVRPJOBN	N	Jobname	Yes
DVRPCLA	N	Class (A, B, O, S, or T)	Yes

Name	Type	Meaning	Report
DVRPDMN	N	Domain number; no longer used	Yes
DVRPPGN	N	Performance group number; no longer used	Yes
DVRPSUSE	N	Percentage of using	Yes
DVRPSDEL	N	Percentage of delay	Yes
DVRPSVCL	N	Service class	Yes
DVRPKIND	N	Device type indicator	Util
DVRPLCUN	N	Logical control unit ID	Util

DI - Tabular report data table ERBDSIT3

RMF builds ERBDSIT3 when using DI as a report type.

Name	Type	Meaning	Report
DSIDTLLN	K	Logical line number	-
DSIDTPSN	K	Sequence number	-
DSI1SID	N	System identifier	Yes
DSI1DATE	N	Starting date	Yes
DSI1TIME	N	Starting time	Yes
DSI1DDNM	N	DD-name	Yes
DSI1DSNM	N	Data set name	Yes
DSI2DATE	N	Ending date	Yes
DSI2TIME	N	Ending time	Yes
DSI2MESS	N	Message field	Yes

DSND - Tabular report data table ERBDNDT3

RMF builds ERBDNDT3 when using DSND as a report type.

Name	Type	Meaning	Report
DNDDTLLN	K	Logical line number	-
DNDDTPSN	K	Sequence number	-
DNDPDSN	N	Data set name	Yes
DNDPVOLU	N	Volume serial	Yes
DNDPJOBN	N	Jobname	Yes
DNDPASID	N	ASID	Yes
DNDPDUSG	N	DUSG (Using %)	Yes
DNDPDDLY	N	DDLY (Delay %)	Yes

DSNJ - Tabular report data table ERBDNJT3

RMF builds ERBDNJT3 when using DSNJ as a report type.

Name	Type	Meaning	Report
DNJDTLLN	K	Logical line number	-
DNJDTPSN	K	Sequence number	-
DNJPASID	N	ASID	Yes

Name	Type	Meaning	Report
DNJPDSN	N	Data set name	Yes
DNJPVOLU	N	Volume	Yes
DNJPDEVN	N	Four-digit device number	Util
DNJPDVN5	N	Five-digit device number	Yes
DNJPDUSG	N	DUSG (Using %)	Yes
DNJPDDLY	N	DDLY (Delay %)	Yes

DSNV - Tabular report data table ERBDNVT3

RMF builds ERBDNVT3 when using DSNV as a report type.

Name	Type	Meaning	Report
DNVDTLLN	K	Logical line number	-
DNVDTPSN	K	Sequence number	-
DNVPDSN	N	Data set name	Yes
DNVPJOBN	N	Jobname	Yes
DNVPASID	N	ASID	Yes
DNVPDUSG	N	DUSG (Using %)	Yes
DNVPDDLY	N	DDLY (Delay %)	Yes

EADM - Tabular report data table ERBSCMT3

RMF builds ERBSCMT3 when using EADM as a report type.

Name	Type	Meaning	Report
SCMDTLLN	K	Logical line number	-
SCMDTPSN	K	Sequence number	-
SCMRPID	N	Card id	Yes
SCMUTL	N	LPAR utilization percentage	Yes
SCMUTLC	N	Total utilization percentage	Yes
SCMDRD	N	LPAR bytes read per second	Yes
SCMDRDC	N	Total bytes read per second	Yes
SCMDWR	N	LPAR bytes written per second	Yes
SCMDWRC	N	Total bytes written per second	Yes
SCMQR	N	LPAR requests processed per second	Yes
SCMQRC	N	Total requests processed per second	Yes
SCMART	N	LPAR response time per request in milliseconds	Yes
SCMARTC	N	Total response time per request in milliseconds	Yes
SCMAQTC	N	Total IOP queue time per request in milliseconds	Yes
SCMTRQ	N	LPAR number of requests	Util
SCMTRQC	N	Total number of requests	Util
SCMHSCR	N	Number of SSCH instructions to all EADM devices per second	Yes
SCMHSCH	N	Total number of SSCH instructions to all EADM devices	Yes
SCMHFPT	N	Function pending time across all EADM devices in milliseconds	Yes

Name	Type	Meaning	Report
SCMHIQT	N	IOP queue time across all EADM devices in milliseconds	Yes
SCMHCRT	N	Command response time across all EADM devices in milliseconds	Yes

Fields in the EADM report header

Name	Type	Meaning	Report
SCMHSCR	N	Number of SSCH instructions to all EADM devices per second	Yes
SCMHSCH	N	Total number of SSCH instructions to all EADM devices	Yes
SCMHFPT	N	Function pending time across all EADM devices in milliseconds	Yes
SCMHIQT	N	IOP queue time across all EADM devices in milliseconds	Yes
SCMHCRT	N	Command response time across all EADM devices in milliseconds	Yes
SCMHRRC	N	Compression request rate (EADM)	Yes
SCMHTPC	N	Compression throughput (EADM)	Yes
SCMHRCC	N	Compression ratio (EADM)	Yes
SCMHRRD	N	Decompression request rate (EADM)	Yes
SCMHTPD	N	Decompression throughput (EADM)	Yes
SCMHRCD	N	Decompression ratio (EADM)	Yes

ENCLAVE - Tabular report data table ERBENCT3

RMF builds ERBENCT3 when using ENCLAVE as a report type.

Name	Type	Meaning	Report
ENCDTLLN	K	Logical line number	-
ENCDTPSN	K	Sequence number	-
ENCENAME	N	Enclave name (generated)	Yes
ENCCLASS	N	Service class	Yes
ENCGOAL	N	Goal time	Yes
ENCGPERC	N	Goal percent	Yes
ENCPER	N	Period	Yes
ENCDENC	N	Dependent enclave indicator	Yes
ENCXENC	N	Multi-system enclave indicator	Yes
ENCTCPU	N	Total CPU time (seconds)	Yes
ENCTIFA	N	Total zAAP time (seconds)	Util
ENCTIFC	N	Total zAAP on CP time (seconds)	Util
ENCDCPU	N	Delta CPU time (seconds)	Pop-Up
ENCDIFA	N	Delta zAAP time (seconds)	Util
ENCDIFC	N	Delta zAAP on CP time (seconds)	Util
ENCDCPUP	N	Delta CPU percentage in Monitor III range	Yes
ENCDIFAP	N	Delta zAAP percentage	Util
ENCDIFCP	N	Delta zAAP on CP percentage	Util
ENCSAMP	N	Total execution samples	Pop-Up
ENCTUSG	N	% Total using samples	Yes
ENCTDLY	N	% Total delay samples	Yes

Name	Type	Meaning	Report
ENCIDLE	N	% Idle	Yes
ENCCUSG	N	% CPU using	Pop-Up
ENCIFAU	N	% zAAP using	Pop-Up
ENCIFCU	N	% zAAP on CP using	Util
ENCCDLY	N	% CPU delay	Pop-Up
ENCIUSG	N	% I/O using	Pop-Up
ENCIDLY	N	% I/O delay	Pop-Up
ENCIFAD	N	% zAAP delay	Pop-Up
ENCCCAP	N	% CPU capping	Pop-Up
ENCSTOR	N	% Storage delay	Pop-Up
ENCUNKN	N	% Unknown	Pop-Up
ENCQUED	N	% Queue delay	Pop-Up
ENCESTYP	N	Subsystem type	Pop-Up
ENCEOWNM	N	Owner name	Pop-Up
ENCEOSYS	N	Owner system	Pop-Up
ENCXTOKN	N	Export token	Pop-Up
ENCATTN	N	Number of attributes in table	Pop-Up
ENCATT00	N	Used to place selected attributes to report, truncated to a length of eight characters	Yes
ENCATT01	N	Accounting Information	Pop-Up
ENCATT02	N	Collection Name	Pop-Up
ENCATT03	N	Connection Type	Pop-Up
ENCATT04	N	Correlation Information	Pop-Up
ENCATT05	N	LU Name	Pop-Up
ENCATT06	N	Net ID	Pop-Up
ENCATT07	N	Package Name	Pop-Up
ENCATT08	N	Plan Name	Pop-Up
ENCATT09	N	Procedure Name	Pop-Up
ENCATT10	N	Process Name	Pop-Up
ENCATT11	N	Scheduling Environment	Pop-Up
ENCATT12	N	Subsystem Collection Name	Pop-Up
ENCATT13	N	Subsystem Instance	Pop-Up
ENCATT14	N	Subsystem Parameter	Pop-Up
ENCATT15	N	Subsystem Type	Pop-Up
ENCATT16	N	Transaction/Job Class	Pop-Up
ENCATT17	N	Transaction/Job Name	Pop-Up
ENCATT18	N	User ID	Pop-Up
ENCATT19	N	Priority	Pop-Up
ENCATT20	N	Client IP Address	Pop-Up
ENCATT21	N	Client User ID	Pop-Up
ENCATT22	N	Client Transaction Name	Pop-Up

Name	Type	Meaning	Report
ENCATT23	N	Client Workstation/Host name	Pop-Up
ENCATT24	N	Client Accounting Information	Pop-Up
ENCTSUP	N	Total ZIIP time (seconds)	Pop-Up
ENCTSUC	N	Total ZIIP on CP time (seconds)	Util
ENCDSUP	N	Delta ZIIP time (seconds)	Pop-Up
ENCDSUC	N	Delta ZIIP on CP time (seconds)	Util
ENCDSUPP	N	Delta ZIIP percentage	Util
ENCDSUCP	N	Delta ZIIP on CP percentage	Util
ENCSUPU	N	% ZIIP using	Pop-Up
ENCSUCU	N	% ZIIP on CP using	Util
ENCSUPD	N	% ZIIP delay	Pop-Up

ENQ - Tabular report data table ERBENQT3

RMF builds ERBENQT3 when using ENQ as a report type.

Name	Type	Meaning	Report
ENQDTLLN	K	Logical line number	-
ENQDTPSN	K	Sequence number	-
ENQPWJOB	N	Jobname of waiting job	Yes
ENQPODEL	N	Overall delay percentage	Yes
ENQPRDEL	N	Percentage of delay for the resource	Yes
ENQPWSTT	N	Status of waiting job	Yes
ENQPMAJS	N	Resource major name and scope or minor name	Yes
ENQPHDEL	N	Holding percentage for the holding job	Yes
ENQPHJOB	N	Jobname of holding job or system name for holding job	Yes
ENQPHSTT	N	Status for the holding job	Yes

ENQR - Tabular report data table ERBEQRT3

RMF builds ERBEQRT3 when using ENQR as a report type.

Name	Type	Meaning	Report
EQRDTLLN	K	Logical line number	-
EQRDTPSN	K	Sequence number	-
EQRPMAJS	N	Resource major name and scope or resource minor name	Yes
EQRPRDEP	N	Percentage of delay for the delayed job	Yes
EQRPWJOB	N	Jobname of delayed job	Yes
EQRPWSTT	N	Status of delayed job	Yes
EQRPHDEP	N	Holding percentage for the holding job	Yes
EQRPHJOB	N	Jobname of holding job or system name	Yes
EQRPHSTT	N	Status of holding job	Yes

HSM - Tabular report data table ERBHSMT3

RMF builds ERBHSMT3 when using HSM as a report type. The table variables are identical to the variables of the ERBJEST3 table; see the ERBJEST3 table for more information.

IOQUEUE - Tabular report data table ERBIOQT3

RMF builds ERBIOQT3 when using IOQUEUE as a report type.

Name	Type	Meaning	Report
IOQDTLLN	K	Logical line number	-
IOQDTPSN	K	Sequence number	-
IOQCPIVC	N	Channel path ID	Yes
IOQPATVC	N	Path attributes	Util
IOQDCMVC	N	DCM-managed channels	Yes
IOQPCUVC	N	Physical CU string	Yes
IOQMMNVC	N	Minimum number of DCM-managed channels used	Yes
IOQMMXVC	N	Maximum number of DCM-managed channels used	Yes
IOQMDFVC	N	Defined number of DCM-managed channels	Yes
IOQLCUVC	N	Logical control unit ID	Yes
IOQCRTVC	N	Contention rate	Yes
IOQDQLVC	N	Delay queue length	Yes
IOQCPTVC	N	Channel path ID taken	Yes
IOQSPBVC	N	Director port busy percent	Yes
IOQCUBVC	N	Control unit busy percent	Yes

JES - Tabular report data table ERBJEST3

RMF builds ERBJEST3 when using JES as a report type.

Name	Type	Meaning	Report
HJSDTLLN	K	Logical line number	-
HJSDTPSN	K	Sequence number	-
HJSPJOB	N	Jobname	Yes
HJSPODEL	N	Overall delay percentage	Yes
HJS1FDEL	N	Delay percentage	Yes
HJS1FCNR	N	Function code	Yes
HJS1EXPL	N	Explanation	Yes
HJS2FDEL	N	Delay percentage	Util
HJS2FCNR	N	Function code	Util
HJS2EXPL	N	Explanation	Util

JOB - Tabular report data table ERBJDJT3

RMF builds ERBJDJT3 when using JOB as a report type.

Name	Type	Meaning	Report
JDJDTLLN	K	Logical line number	-
JDJDTPSN	K	Sequence number	-

Name	Type	Meaning	Report
JDJLDAN	N	Jobname or summary	Yes
JDJLASID	N	Address space identification	Yes
JDJCLASS	N	Class (A, B, E, O, S, or T)	Util
JDJCLASX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
JDJLDMN	N	Domain number; no longer used	Yes
JDJLPGN	N	Performance group number; no longer used	Yes
JDJPSVCL	N	Service class name	Yes
JDJLWFL	N	Work flow percentage	Yes
JDJLUSP	N	Processor using percentage	Yes
JDJLUSD	N	Device using percentage	Yes
JDJLUSG	N	Using percentage	Util
JDJLDEL	N	Delay percentage	Yes
JDJLIDL	N	Idle percentage	Yes
JDJLUKN	N	Unknown percentage	Yes
JDJLPROC	N	Processor delay percentage	Yes
JDJLDEV	N	Device delay percentage	Yes
JDJLSTOR	N	Storage delay percentage	Yes
JDJLSUBS	N	SUBS delay percentage	Yes
JDJLOPER	N	Operator delay percentage	Yes
JDJLENQ	N	ENQ delay percentage	Yes
JDJLJES	N	JES delay percentage	Util
JDJLHSM	N	HSM delay percentage	Util
JDJLXCF	N	XCF delay percentage	Util
JDJLMNT	N	Operator mount delay percentage	Util
JDJLMES	N	Operator message delay percentage	Util
JDJLQUI	N	Operator quiesce delay percentage	Util
JDJLREAS	N	Primary reason	Yes

LOCKSP - Tabular report data table ERBLSPT3

RMF builds ERBLSPT3 when using LOCKSP as a report type.

Name	Type	Meaning	Report
LSPDTLLN	K	Logical line number	-
LSPDTPSN	K	Sequence number	-
LSPPRES	N	Resource name	Yes
LSPPJT	N	Spinner jobname / spin lock type	Yes
LSPPAC	N	Spinner address space ID / CPU ID	Yes
LSPPRAD	N	Spinner request address	Yes
LSPPHELD	N	Spin lock held percentage	Yes
LSPPSPIN	N	Spin percentage	Yes

LOCKSU - Tabular report data table ERBLSUT3

RMF builds ERBLSUT3 when using LOCKSU as a report type.

Name	Type	Meaning	Report
LSUDTLLN	K	Logical line number	-
LSUDTPSN	K	Sequence number	-
LSUPRES	N	Resource name	Yes
LSUPTYPE	N	Lock Type	Yes
LSUPJOB	N	Holder job name	Yes
LSUPASI	N	Holder address space ID	Yes
LSUPRAD	N	Request address	Yes
LSUPHELD	N	Holding percentage	Yes
LSUPINTR	N	Holding while interrupted percentage	Yes
LSUPDISP	N	Holding while dispatched percentage	Yes
LSUPSUSP	N	Holding while suspended percentage	Yes

OPD - Tabular report data table ERBOPDT3

RMF builds ERBOPDT3 when using OPD as a report type.

Name	Type	Meaning	Report
OPDDTLLN	K	Logical line number	-
OPDDTPSN	K	Sequence number	-
OPDPJOBN	N	Jobname	Yes
OPDPUSEN	N	User name	Yes
OPDPASID	N	ASID	Yes
OPDPASIX	N	Hexadecimal ASID	Yes
OPDPPRID	N	Process ID	Yes
OPDPPPID	N	Parent's process ID	Yes
OPDPLATW	N	Waiting for process latch	Yes
OPDPSTAT	N	Process state	Yes
OPDPAPPL	N	Percentage of TCB and SRB time	Yes
OPDPTOT	N	Total computing time since process has been started	Yes
OPDPSERV	N	Server type	Yes
OPDIPRID	N	Process ID	Pop-Up
OPDIPPID	N	Parent's process ID	Pop-Up
OPDIJOBN	N	Jobname	Pop-Up
OPDIUSEN	N	User name	Pop-Up
OPDIASID	N	ASID	Pop-Up
OPDIASIX	N	Hexadecimal ASID	Pop-Up
OPDITIMD	N	Start time/date	Pop-Up
OPDIAPPL	N	Percentage of TCB and SRB time	Pop-Up
OPDITOT	N	Total computing time since process has been started	Pop-Up
OPDILPID	N	Latch process ID the process is waiting for (0 = not waiting)	Pop-Up

Name	Type	Meaning	Report
OPDICMD	N	Command buffer	Pop-Up
OPDISERN	N	Server name	Pop-Up
OPDISERV	N	Server type	Pop-Up
OPDIACTF	N	Number of active files	Pop-Up
OPDIMAXF	N	Maximum files	Pop-Up
OPDISTAT	N	Process state	Pop-Up
OPDISTA1	N	1. possible state	Pop-Up
OPDISTA2	N	2. possible state	Pop-Up
OPDISTA3	N	3. possible state	Pop-Up

PCIE – Tabular report data table ERBPCIT3

RMF builds ERBPCIT3 when using PCIE as a report type.

Name	Type	Meaning	Report
PCIDTLLN	K	Logical line number	-
PCIDTPSN	K	Sequence number	-
PCIEPFID	N	PCIE function id	Yes
PCIESTAT	N	Function status	Yes
PCIEPCID	N	Physical channel id	Yes
PCIEDEVT	N	Device type	Yes
PCIEALLT	N	Function alloc time %	Yes
PCIEJOBN	N	Owning job name	Yes
PCIEASID	N	Owning ASID	Yes
PCIELOOP	N	PCI Load operations rate (RoCE, ISM function, Hardware Accelerator)	Yes
PCIESTOP	N	PCI Store operations rate (RoCE, ISM function, Hardware Accelerator)	Yes
PCIESBOP	N	PCI Store Block operations rate (RoCE, ISM function, Hardware Accelerator)	Yes
PCIERFOP	N	PCI Refresh operations rate (RoCE, ISM function, Hardware Accelerator)	Yes
PCIEDMAR	N	Transfer read rate (Synchronous I/O, RoCE, Hardware Accelerator)	Yes, Pop-Up
PCIEDMAW	N	Transfer write rate	Yes, Pop-Up
PCIEDPKR	N	Received packets rate (RoCE, Synchronous I/O)	Pop-Up
PCIEDPKT	N	Transmitted packets rate (RoCE, Synchronous I/O)	Pop-Up
PCIEDWUP	N	Work unit rate (Hardware Accelerator)	Pop-Up
PCIEDAUT	N	Adapter utilization (Hardware Accelerator)	Pop-Up
PCIEADAT	N	Allocation date	Pop-Up
PCIEATIM	N	Allocation time	Pop-Up
PCIEDMAN	N	Number of DMA address spaces	Util
PCIEFTYP	N	Hardware Accelerator application type	Util
PCIEFBSY	N	Time busy % (Hardware Accelerator, Synchronous I/O)	Pop-Up
PCIEFTR	N	Hardware Accelerator transfer rate	Util
PCIEFRET	N	Request execution time (Hardware Accelerator)	Pop-Up
PCIEFRES	N	Request execution time standard deviation (Hardware Accelerator)	Pop-Up

Name	Type	Meaning	Report
PCIEFRQT	N	Request queue time (Hardware Accelerator)	Pop-Up
PCIEFRQS	N	Request queue time standard deviation (Hardware Accelerator)	Pop-Up
PCIEFRSZ	N	Request size (Hardware Accelerator)	Pop-Up
PCIE1RRC	N	Application request rate compression (Hardware Accelerator)	Pop-Up
PCIE1TPC	N	Application throughput compression (Hardware Accelerator)	Pop-Up
PCIE1RCC	N	Application ratio compression (Hardware Accelerator)	Pop-Up
PCIE1RRD	N	Application request rate de-compression (Hardware Accelerator)	Pop-Up
PCIE1TPD	N	Application throughput de-compression (Hardware Accelerator)	Pop-Up
PCIE1RCD	N	Application ratio de-compression (Hardware Accelerator)	Pop-Up
PCIE1BPS	N	Buffer pool memory size (Hardware Accelerator)	Pop-Up
PCIE1BPU	N	Buffer pool utilization % (Hardware Accelerator)	Pop-Up
PCIENET1	N	Physical-network identifier (PNET ID) that identifies the first port of the RoCE device or ISM function	Pop-Up
PCIENET2	N	Physical-network identifier (PNET ID) that identifies the second port of the RoCE device	Pop-Up
PCIEPOID	N	Port ID (RoCE, ISM function, Synchronous I/O)	Pop-Up
PCIESERN	N	Serial number of the storage controller (Synchronous I/O)	Pop-Up
PCIETYMO	N	Type and model of the storage controller (Synchronous I/O)	Pop-Up
PCIETBPC	N	Time busy % (CPC) (Synchronous I/O)	Pop-Up
PCIERRT	N	Request rate (Synchronous I/O)	Pop-Up
PCIERRTC	N	Request rate (CPC) (Synchronous I/O)	Pop-Up
PCIETRRC	N	Transfer read rate (CPC) (Synchronous I/O)	Pop-Up
PCIEXRR	N	Transfer read ratio (Synchronous I/O)	Pop-Up
PCIEXRRC	N	Transfer read ratio (CPC) (Synchronous I/O)	Pop-Up
PCIETWRC	N	Transfer write rate (CPC) (Synchronous I/O)	Pop-Up
PCIEXWR	N	Transfer write ratio (Synchronous I/O)	Pop-Up
PCIEXWRC	N	Transfer write ratio (CPC) (Synchronous I/O)	Pop-Up
PCIESRR	N	Successful request % (Synchronous I/O)	Pop-Up
PCIESRRC	N	Successful request % (CPC) (Synchronous I/O)	Pop-Up
PCIELKID	N	The identifier of the synchronous I/O link that is configured in the storage controller.	Pop-Up

PROC - Tabular report data table ERBPRCT3

RMF builds ERBPRCT3 when using PROC as a report type.

Name	Type	Meaning	Report
PRCDTLLN	K	Logical line number	-
PRCDTPSN	K	Sequence number	-
PRCPJOB	N	Jobname	Yes
PRCPASI	N	Address space ID of the job (decimal format)	Util
PRCPCLA	N	Class (A, B, E, O, S, or T)	Util
PRCPCLAX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
PRCPDMN	N	Domain number; no longer used	Yes

Name	Type	Meaning	Report
PRCPPGN	N	Performance group number; no longer used	Yes
PRCPSVCL	N	Service class name	Yes
PRCPODEL	N	Overall delay percentage for this address space.	Util
PRCPOUSE	N	Overall using percentage for this address space.	Util
PRCPTYPE	N	Processor type	Yes
PRCPTST	N	Overall application percentage for this address space.	Util
PRCPCAP	N	Capping delay percentage	Util
PRCPETST	N	Overall application percentage including EAppl percentage	Util
PRCPAPPL	N	Overall application percentage on behalf of this address space and processor type	Util
PRCPEAPP	N	Overall application percentage including EAppl percentage on behalf of this address space and processor type	Yes
PRCPTWFL	N	Overall workflow percentage of this address space and processor type	Util
PRCPTDEL	N	Overall delay percentage for this address space and processor type	Yes
PRCPTUSE	N	Overall using percentage for this address space and processor type	Yes
PRCPAACP	N	% zAAP on CP using	Util
PRCPIICP	N	% ZIIP on CP using	Util
PRC1SDEL	N	Delay percentage caused by jobname1	Yes
PRC1JOBN	N	Jobname1	Yes
PRC2SDEL	N	Delay percentage caused by jobname2	Yes
PRC2JOBN	N	Jobname2	Yes
PRC3SDEL	N	Delay percentage caused by jobname3	Yes
PRC3JOBN	N	Jobname3	Yes
PRCTCPUT	N	Total CPU time (milliseconds)	Util

PROCU - Tabular report data table ERBPRUT3

RMF builds ERBPRUT3 when using PROCU as a report type.

Name	Type	Meaning	Report
PRUDTLLN	K	Logical line number	-
PRUDTPSN	K	Sequence number	-
PRUPJOB	N	Jobname	Yes
PRUPASI	N	Address space ID of the job (decimal format)	Yes
PRUPCLA	N	Class (A, B, E, O, S, or T)	Util
PRUPCLAX	N	Class (A, B, E, O, S, or T) with possible extension O	Yes
PRUPSVCL	N	Service class name	Yes
PRUPCLP	N	Service class period	Util
PRUPCPT	N	Time on CP %	Yes
PRUPAACT	N	zAAP time on CP %	Util
PRUPIICT	N	ZIIP time on CP %	Yes
PRUPCPE	N	CP EAppl %	Yes
PRUPAAPE	N	zAAP EAppl %	Util

Name	Type	Meaning	Report
PRUPIIPE	N	ZIIP EAppl %	Yes
PRUPTOTC	N	Percentage of total accumulated CPU time as sum of TCB time, global and local SRB time and preemptable or client SRB time, consumed on behalf of this address space.	Util
PRUPTOTE	N	Percentage of total accumulated CPU time as sum of TCB time, global and local SRB time, preemptable or client SRB time, and enclave time consumed within this address space.	Util
PRUPTCB	N	Percentage of TCB time consumed in this address space.	Util
PRUPSRB	N	Percentage of SRB time consumed in this address space by local or global SRBs.	Util
PRUPPCS	N	Percentage of preemptable or client SRB time consumed on behalf of this address space.	Util
PRUPEPS	N	Percentage of preemptable or client SRB and enclave CPU time consumed within this address space.	Util
PRUTCPUT	N	Total CPU time (milliseconds)	Util

RLSDS - Tabular report data table ERBVRDT3

RMF builds ERBVRDT3 when using RLSDS as a report type.

Name	T	Description of the variable	Report
VRDDTLLN	K	Logical line number	-
VRDDTPSN	K	Sequence number	-
VRDPNAM	N	VSAM sphere name, Data set name, MVS system name, Access type, Response time, Read rate, Read BMF hit percentage, Read CF hit percentage	Yes
VRDPRDAS	N	Read DASD hit percentage	Yes
VRDPBMFV	N	BMF valid percentage	Yes
VRDPBMFF	N	BMF false invalid percentage	Yes
VRDPWRTE	N	Write rate	Yes
VRDPCALO	N	Castout lock percentage	Util
VRDPREDA	N	Redo activity percentage	Util
VRDPRRED	N	Recursive redo percentage	Util
VRDPIND	N	Report indication	Util

RLSLRU - Tabular report data table ERBVRLT3

RMF builds ERBVRLT3 when using RLSLRU as a report type.

Name	T	Description of the variable	Report
VRLDTLLN	K	Logical line number	-
VRLDTPSN	K	Sequence number	-
VRLPSYS	N	MVS system name	Yes
VRLPACT	N	Average CPU time	Yes
VRLPBSG	N	Buffer size goal	Yes
VRLPBSH	N	Buffer size high	Yes
VRLPBSO	N	Buffer percentage accelerated	Yes
VRLPBSS	N	Buffer percentage reclaiming	Yes

Name	T	Description of the variable	Report
VRLPABMF	N	Average BMF hit percentage	Yes
VRLPACAC	N	Average Cache hit percentage	Yes
VRLPADAS	N	Average DASD hit percentage	Yes
VRLPCALO	N	Castout lock percentage	Util
VRLPREDA	N	Redo activity percentage	Util
VRLPRRED	N	Recursive redo percentage	Util
VRLISYS	N	MVS system name	Pop-Up
VRLIFPL	N	Fixed pages low	Pop-Up
VRLIFPH	N	Fixed pages high	Pop-Up
VRLIFPA	N	Fixed pages average	Pop-Up
VRLIFIXS	N	Fixed storage	Pop-Up
VRLIRSP	N	Real storage %	Pop-Up
VRLILSn	N	Buffer count by pool of size nK, low value, below 2 GB, where n is 2, 4, ..., 32,	Pop-Up
VRLIHSn	N	Buffer count by pool of size nK, high value, below 2 GB, where n is 2, 4, ..., 32,	Pop-Up
VRLICSn	N	Buffer count by pool of size nK, average value, below 2 GB, where n is 2, 4, ..., 32	Pop-Up
VRLAFPL	N	Fixed pages low above 2 GB	Pop-Up
VRLAFPH	N	Fixed pages high above 2 GB	Pop-Up
VRLAFPA	N	Fixed pages average above 2 GB	Pop-Up
VRLAFIXS	N	Fixed storage above 2 GB	Pop-Up
VRLARSP	N	Real storage % above 2 GB	Pop-Up
VRLALSn	N	Buffer count by pool of size nK, low value, above 2 GB, where n is 2, 4, ..., 32,	Pop-Up
VRLAHSn	N	Buffer count by pool of size nK, high value, above 2 GB, where n is 2, 4, ..., 32,	Pop-Up
VRLACSn	N	Buffer count by pool of size nK, average value, above 2 GB, where n is 2, 4, ..., 32,	Pop-Up

RLSSC - Tabular report data table ERBVRST3

RMF builds ERBVRST3 when using RLSSC as a report type.

Name	T	Description of the variable	Report
VRSDTLLN	K	Logical line number	-
VRSDTPSN	K	Sequence number	-
VRSPNAM	N	Storage class name, MVS system name, CF structure name, Access type	Yes
VRSPRTIM	N	Average response time	Yes
VRSPRRTE	N	Read rate	Yes
VRSPRBMF	N	Read BMF hit percentage	Yes
VRSPRCF	N	Read CF hit percentage	Yes
VRSPRDAS	N	Read DASD hit percentage	Yes
VRSPBMFV	N	BMF valid percentage	Yes
VRSPBMFF	N	BMF false invalid percentage	Yes

Name	T	Description of the variable	Report
VRSPWRTE	N	Write rate	Yes
VRSPIND	N	Report indication	Util

SPACED - Tabular report data table ERBSPDT3

RMF builds ERBSPDT3 when using SPACED as a report type.

Name	T	Description of the variable	Report
SPDDTLLN	K	Logical line number	-
SPDDTPSN	K	Sequence number	-
SPDPVOL	N	Volume	Yes
SPDPTSP	N	Total capacity in MB	Yes
SPDPFSP	N	Free space in MB	Yes
SPDPFSR	N	Free space percentage	Yes
SPDPLBK	N	Largest block in MB	Yes
SPDPSGN	N	Storage group name	Yes

SPACEG - Tabular report data table ERBSPGT3

RMF builds ERBSPGT3 when using SPACEG as a report type.

Name	T	Description of the variable	Report
SPGDTLLN	K	Logical line number	-
SPGDTPSN	K	Sequence number	-
SPGPSGN	N	Storage Group	Yes
SPGPTSP	N	Total capacity in MB	Yes
SPGPFSP	N	Free space in MB	Yes
SPGPFSR	N	Free space percentage	Yes
SPGPNVO	N	Number of Volumes	Yes

STOR - Tabular report data table ERBSTRT3

RMF builds ERBSTRT3 when using STOR as a report type.

Name	T	Description of the variable	Report
STRDTLLN	K	Logical line number	-
STRDTPSN	K	Sequence number	-
STRPJOB	N	Jobname	Yes
STRPCLA	N	Class (A, B, O, S, or T)	Yes
STRPDMN	N	Domain number; no longer used	Yes
STRPPGN	N	Performance group number; no longer used	Yes
STRPSVCL	N	Service class name	Yes
STRPODEL	N	Overall delay percentage	Yes
STR1SDEL	N	Delay percentage COMM	Yes
STR2SDEL	N	Delay percentage local	Yes
STR3SDEL	N	Delay percentage VIO	Util

Name	T	Description of the variable	Report
STR4SDEL	N	Delay percentage SWAP	Yes
STR5SDEL	N	Delay percentage OUTR	Yes
STR6SDEL	N	Cross memory delay %	Util
STR7SDEL	N	Hiperspace delay %	Util
STR8SDEL	N	Other delays % (including VIO, XMEM and HIPR)	Yes
STRPACTV	N	Average ACTV frames	Util
STRPFIXD	N	Average fixed frames total	Util
STRPIDLE	N	Average IDLE frames	Util
STRPWSET	N	Average working set frames	Yes
STRPWSEX	N	Average ES working set frames	Yes

STORC - Tabular report data table ERBCSUT3

RMF builds ERBCSUT3 when using STORC as a report type.

Name	T	Description of the variable	Report
CSUDTLLN	K	Logical line number	-
CSUDTPSN	K	Sequence number	-
CSXNAME	N	Jobname	Yes
CSXACT	N	Active column	Yes
CSXCLA	N	Class (A, B, O, S, or T)	Yes
CSXDMN	N	Domain number; no longer used	Yes
CSXPGN	N	Performance group number; no longer used	Yes
CSXCSN	N	Service class name	Yes
CSXASID	N	Address space identifier	Yes
CSXTIME	N	Elapsed time	Yes
CSXPCSA	N	Percentage of CSA and RUCSA	Yes
CSXPECS	N	Percentage of ECSA and ERUCSA	Yes
CSXPSQA	N	Percentage of SQA	Yes
CSXPESQ	N	Percentage of ESQA	Yes
CSXACSA	N	Amount of CSA and RUCSA	Yes
CSXAECS	N	Amount of ECSA and ERUCSA	Yes
CSXASQA	N	Amount of SQA	Yes
CSXAESQ	N	Amount of ESQA	Yes
CSXJESID	N	JES identifier	Util
CSXTDATE	N	Termination date	Util
CSXTTIME	N	Termination time	Util

STORCR - Tabular report data table ERBCRST3

RMF builds ERBCRST3 when using STORCR as a report type.

Name	T	Description of the variable	Report
CSUDTLLN	K	Logical line number	-

Name	T	Description of the variable	Report
CSUDTPSN	K	Sequence number	-
CSXNAME	N	Jobname	Yes
CSXJESID	N	JES identifier	Yes
CSXTDATE	N	Termination date	Yes
CSXTTIME	N	Termination time	Yes
CSXACSA	N	Amount of CSA and RUCSA	Yes
CXSAECS	N	Amount of ECSA and ERUCSA	Yes
CSXASQA	N	Amount of SQA	Yes
CSXAESQ	N	Amount of ESQA	Yes

STORF - Tabular report data table ERBSTFT3

RMF builds ERBSTFT3 when using STORF as a report type.

Name	Type	Meaning	Report
STFDTLLN	K	Logical line number	-
STFDTPSN	K	Sequence number	-
STFPJOB	N	Jobname	Yes
STFPCLA	N	Class (A, B, O, S, or T)	Yes
STFPDMN	N	Domain number; no longer used	Yes
STFPPGN	N	Performance group number; no longer used	Yes
STFPSVCL	N	Service class name	Yes
STFPTOTL	N	Frame occupancy TOTAL	Yes
STFPACTV	N	Frame occupancy ACTV	Yes
STFPIDLE	N	Frame occupancy IDLE	Yes
STFPWSET	N	Active frames WSET	Yes
STFPFIXD	N	Active frames FIXED	Yes
STFPDIV	N	Active frames DIV	Yes
STFPAUXS	N	Auxiliary storage slots	Yes
STFPPGIN	N	Page-in Rate	Yes
STFPEXIN	N	Page-in rate from expanded storage	Util
STFPSPPI	N	Shared pages page-in rate from auxiliary storage	Util
STFPTOTS	N	Total number of shared page views	Util
STFPSVIN	N	Total number of valid shared pages	Util
STFPSPVL	N	Shared pages validation rate	Util
STFPLMO	N	Number of fixed 1 MB memory objects allocated	Util
STFPLPR	N	Number of 1 MB frames fixed in real storage	Util
STFPFREM	N	Number of freemained frames	Util
STFPGMO	N	Number of 2 GB memory objects allocated	Util
STFPGPR	N	Number of 2 GB frames fixed in central storage	Util

STORM - Tabular report data table ERBSTMT3

RMF builds ERBSTMT3 when using STORM as a report type.

Name	Type	Meaning	Report
STMDTLLN	K	Logical line number	-
STMDTPSN	K	Sequence Number	-
STMPJOB	N	Jobname	Yes
STMPASI	N	Address space identifier	Yes
STMPCLA	N	Class (A, B, O, S, or T)	Yes
STMPSVCL	N	Service class name	Yes
STMPCLP	N	Service class period	Util
STMPTMO	N	Average number of memory objects allocated (by this address space)	Yes
STMPCMO	N	Average number of high virtual common memory objects allocated (by this address space)	Yes
STMPSMO	N	Average number of high virtual shared memory objects allocated (by this address space)	Yes
STMPPMO	N	Average number of high virtual private memory objects allocated (by this address space)	Util
STMPLMO	N	Average number of fixed memory objects allocated with this address space as the owner that can be backed in 1 MB frames	Util
STMPLSMO	N	Average number of shared memory objects allocated with this address space as the owner that can be backed in 1 MB frames	Util
STMPLFF	N	Average number of 1 MB page-fixed frames that are used by pageable/DREF memory objects owned by this address space	Yes
STMPLPF	N	Average number of 1 MB frames that are used by pageable/DREF memory objects owned by this address space.	Yes
STMPLFR	N	Average number of fixed 1 MB pages backed in central storage owned by this address space	Yes
STMPVTB	N	Average amount of storage allocated from high virtual memory in memory objects owned by this address space	Yes
STMPCMB	N	Average amount of high virtual common storage allocated owned by this address space	Yes
STMPVSB	N	Average amount of storage allocated in shared memory objects owned by this address space	Yes
STMPPMB	N	Average amount of high virtual private storage allocated owned by this address space	Util
STMPHCB	N	High water mark for the amount of high virtual common storage allocated (in bytes)	Util
STMPHSB	N	High water mark for the amount of storage allocated in shared memory objects (in bytes)	Util
STMPLMB	N	Address space memory limit	Util
STMPGMO	N	Average number of fixed memory objects allocated with this address space as the owner that are backed in 2 GB frames	Util
STMPGFR	N	Average number of fixed 2 GB pages backed in central storage owned by this address space	Yes

Fields in the STORM report header

Name	T	Description of the variable	Report
STMHSMO	N	Average number of high virtual shared memory objects allocated in the system	Yes
STMHCMO	N	Average number of high virtual common memory objects allocated in the system	Yes
STMHSFR	N	Average number of high virtual shared memory pages backed in central storage	Yes
STMHSSIZ	N	Percentage of high virtual shared memory in use by the system	Util

Name	T	Description of the variable	Report
STMHCFR	N	Average number of high virtual common memory pages backed in central storage	Yes
STMHCSIZ	N	Percentage of high virtual common memory in use by the system	Yes
STMHCFFR	N	Average number of high virtual common memory pages fixed in central storage	Util
STMHSASL	N	Average number of high virtual shared memory auxiliary storage slots (DASD and SCM)	Util
STMHCASL	N	Average number of high virtual common memory auxiliary storage slots (DASD and SCM)	Util
STMHLMO	N	Average number of fixed memory objects that are allocated in the system and can be backed in 1 MB frames	Yes
STMHLCMO	N	Average number of fixed memory objects allocated in high virtual common storage that can be backed in 1 MB frames	Util
STMHLCMU	N	Average number of fixed memory objects allocated in high virtual common storage that can be backed in 1 MB frames whose owner is no longer active	Util
STMHLSMO	N	Average number of memory objects allocated in high virtual shared storage that can be backed in 1 MB frames	Util
STMHLFR	N	Average number of 1 MB pages fixed in central storage	Util
STMHLSIZ	N	Percentage of the maximum number of 1 MB frames that are used by fixed 1 MB pages.	Yes
STMHLFF	N	Average maximum number of 1 MB frames that can be used by fixed 1 MB pages.	Yes
STMHLF4K	N	No longer used.	Util
STMHLCFR	N	Average number of 1 MB high virtual common memory pages backed in central storage	Yes
STMHLCPU	N	Average number of 1 MB high virtual common memory pages whose owner is no longer active	Util
STMHFSIZ	N	Percentage of 1 MB frames that are in-use and are no longer available for fixed 1 MB pages.	Util
STMHLPF	N	No longer used.	Util
STMHLP4K	N	No longer used.	Util
STMHLFPF	N	No longer used.	Util
STMHLPFR	N	Average number of failed 1 MB pageable pages that were requested	Util
STMHLPFC	N	Average number of demoted 1 MB pageable pages that were converted from 1 MB pages to 4K pages	Util
STMHPSIZ	N	Percentage of 1 MB frames used by pageable/DREF memory objects.	Yes
STMHGMO	N	Average number of fixed memory objects that are allocated in the system and are backed in 2 GB frames	Yes
STMHGFR	N	Average number of 2 GB pages fixed in central storage	Util
STMHGFF	N	Average number of 2 GB frames that can be used by fixed 2 GB memory objects	Yes
STMHGSIZ	N	Percentage of the maximum number of 2 GB frames that are used by fixed 2 GB memory objects.	Yes
STMHUSIZ	N	Percentage of 1 MB frames in central storage that are in-use.	Yes
STMHLF	N	Total number of 1 MB frames in central storage.	Yes

STORR - Tabular report data table ERBSRRT3

RMF builds ERBSRRT3 when using STORR as a report type.

Name	T	Description of the variable	Report
SRRDTLLN	K	Logical line number	-

Name	T	Description of the variable	Report
SRRDTPSN	K	Sequence number	-
SRRVOLVC	N	Volume serial number	Yes
SRRDEVTY	N	Device type	Yes
SRRCUTY	N	Control unit type	Yes
SRREXPCT	N	Number of exposures	Yes
SRRUSVC	N	Percentage of using	Util
SRRA1VC	N	Percentage of active	Yes
SRRA2VC	N	Percentage of connect	Yes
SRRA3VC	N	Percentage of disconnect	Yes
SRRA4VC	N	Percentage of pending	Yes
SRRA5VC	N	Percentage of DLY-DB	Util
SRRA6VC	N	Percentage of DLY-CUB	Util
SRRA7VC	N	Percentage of DLY-SPB	Util
SRRSPTVC	N	Space type	Yes
SRRAUTOT	N	Average active users: TOTAL	Yes
SRRAULOC	N	Average active users: LOCAL	Yes
SRRAUSWP	N	Average active users: SWAP	Yes
SRRAUCOM	N	Average active users: COMM	Yes
SRRPDLYR	N	Delay type header	Yes
SRRPDLYP	N	Delay reason percentage	Util

STORS - Tabular report data table ERBSRST3

RMF builds ERBSRST3 when using STORS as a report type.

Name	T	Description of the variable	Report
SRSDTLLN	K	Logical line number	-
SRSDTPSN	K	Sequence number	-
SRSPDMPG	N	WLM group name for graphic report	Yes
SRSPDMN	N	Domain number; no longer used	Yes
SRSPPGN	N	Performance group number; no longer used	Yes
SRSPGNAM	N	Name of WLM group	Yes
SRSPGTYP	N	Type of WLM group	Yes
SRSPTOTU	N	Total number of users	Yes
SRSPACTU	N	Number of active users	Yes
SRS1SDEL	N	Average number delayed for ANY	Yes
SRS2SDEL	N	Average number delayed for COMM	Yes
SRS3SDEL	N	Average number delayed for LOCL	Yes
SRS4SDEL	N	Average number delayed for VIO	Util
SRS5SDEL	N	Average number delayed for SWAP	Yes
SRS6SDEL	N	Average number delayed for OUTR	Yes
SRS7SDEL	N	Average number delayed for cross memory	Util

Name	T	Description of the variable	Report
SRS8SDEL	N	Average number delayed for hiperspace	Util
SRS9SDEL	N	Average number delayed for other reasons, including VIO, XMEM and HIPR	Yes
SRSPACTV	N	Average ACTV frames	Yes
SRSPFIXD	N	Average FIXED frames	Yes
SRSPIDLE	N	Average IDLE frames	Yes
SRSPPGIN	N	Page-in rate	Yes
SRSRCTNT	N	Report class is a tenant report class	Util

SYSENQ - Tabular report data table ERBEQST3

RMF builds ERBEQST3 when using SYSENQ as a report type.

Name	T	Description of the variable	Report
EQSDTLLN	K	Logical line number	-
EQSDTPSN	K	Sequence number	-
EQSPMAJN	N	Resource major name or resource minor name	Yes
EQSPWDEP	N	Percentage of delay for the delayed job	Yes
EQSPWJOB	N	Jobname of delayed job	Yes
EQSPWSNM	N	MVS system name of delayed job	Yes
EQSPWSTT	N	Status of delayed job	Yes
EQSPHUSP	N	Holding percentage for the holding job	Yes
EQSPHJOB	N	Jobname of holding job	Yes
EQSPHSNM	N	MVS system name of holding job	Yes
EQSPHSTT	N	Status of holding job	Yes

SYSINFO - Tabular report data table ERBSYST3

RMF builds ERBSYST3 when using SYSINFO as a report type.

Name	Type	Meaning	Report
SYSDTLLN	K	Logical line number	-
SYSDTPSN	K	Sequence number	-
SYSNAMVC	N	WLM group name	Yes
SYSTYPVC	N	Type of WLM group	Yes
SYSWFLVC	N	Workflow percentage	Yes
SYSTUSVC	N	Average number of total users	Yes
SYSAUSVC	N	Average number of active users	Yes
SYSTRSVC	N	Transactions / sec	Yes
SYSAFCVC	N	Active frames percentage	Util
SYSVECVC	N	Vector utilization	Util
SYSRSPM	N	Average response time per transaction in milliseconds	Yes
SYSAUPVC	N	Average number using PROC	Yes
SYSAUDVC	N	Average number using DEV	Yes
SYSADPVC	N	Average number delayed for PROC	Yes

Name	Type	Meaning	Report
SYSADDVC	N	Average number delayed for DEV	Yes
SYSADSVC	N	Average number delayed for STOR	Yes
SYSADUVC	N	Average number delayed for SUBS	Yes
SYSADOVC	N	Average number delayed for OPER	Yes
SYSADEVC	N	Average number delayed for ENQ	Yes
SYSADJVC	N	Average number delayed for JES	Util
SYSADHVC	N	Average number delayed for HSM	Util
SYSADXVC	N	Average number delayed for XCF	Util
SYSADNVC	N	Average number delayed for Mount	Util
SYSADMVC	N	Average number delayed for Message	Util
SYSCPUVC	N	Percentage of the maximum general purpose processor capacity spent on behalf of a group/class	Util
SYSSRBVC	N	Percentage of the maximum general purpose processor capacity spent by SRB work on behalf of a group/class	Util
SYSTCBVC	N	Percentage of the maximum general purpose processor capacity used by non-enclave TCB work that executed within a class or group	Util
SYSEAPVC	N	Percentage of the maximum general purpose processor capacity consumed within a class or group (including enclave time)	Util
SYSIFAVC	N	Percentage of the maximum zAAP processor capacity used within a class or group	Util
SYSCPVC	N	Percentage of the maximum general purpose processor capacity used by non-enclave TCB work that executed within a class or group	Util
SYSIFCVC	N	Percentage of the maximum general purpose processor capacity used by zAAP eligible work that executed within a class or group	Util
SYSRSPVC	N	Average response time per transaction in seconds	Util
SYSVELVC	N	Execution velocity	Util
SYSUGMVC	N	% using	Util
SYSUGPVC	N	% using processor	Util
SYSUGDVC	N	% using device	Util
SYSWGDVC	N	% device workflow	Util
SYSWGPVC	N	% processor workflow	Util
SYSDGMVC	N	% delay	Util
SYSUJMVC	N	Average number users using	Util
SYSDJMVC	N	Average number users delayed	Util
SYSDGEVC	N	% delay for ENQ	Util
SYSDGHVC	N	% delay for HSM	Util
SYSDGDVC	N	% delay for DEV	Util
SYSDGJVC	N	% delay for JES	Util
SYSDGOVC	N	% delay for OPER	Util
SYSDGPVC	N	% delay for PROC	Util
SYSDGSVC	N	% delay for STOR	Util
SYSDGUVC	N	% delay for SUBS	Util
SYSDGXVC	N	% delay for XCF	Util

Name	Type	Meaning	Report
SYSSUPVC	N	Percentage of the maximum zIIP processor capacity used within a class or group	Util
SYSSUCVC	N	Percentage of the maximum general purpose processor capacity used by zIIP eligible work that executed within a class or group	Util
SYSPDPVC	N	CPU time in seconds, that transactions of a class or group were running at a promoted dispatching priority during the report interval.	Util
SYSTODVC	N	% of total delay samples	Util
SYSCPDVC	N	% of CP delay samples	Util
SYSAPDVC	N	% of ZAAP delay samples	Util
SYSIPDVC	N	% of ZIIP delay samples	Util
SYSRGCVC	N	CPU capping because resource group maximum being enforced	Util
SYSRCTNT	N	Report class is a tenant report class	Util
SYSMEMUS	N	Memory used, actual	Util

Fields in the SYSINFO report header

Name	Description of the variable	Report
SYSPARVC	Partition name	Yes
SYSMODVC	Processor family	Yes
SYSMDLVC	Processor model	Yes
SYSTSVVC	Appl% CP	Yes
SYSPOLVC	Policy name	Yes
SYSPRVVC	Average number of online CPs	Yes
SYSCUVVC	Average CPU Util% of CPs	Yes
SYSTSEVC	EAppl% CP	Yes
SYSPADVC	Policy date	Yes
SYSPRIVC	Average number of online logical processors / threads of type zAAP	Util
SYSLCPVC	Average MVS Util% of CPs	Yes
SYSAPIVC	Appl% zAAP	Util
SYSPATVC	Policy time	Yes
SYSPRTVC	Average number of online logical processors / threads of type zIIP	Yes
SYSAPTVC	Appl% ZIIP	Yes
SYSCVAVC	Whether CPU reconfiguration changes occurred during the reporting interval (YES or NO).	Util
SYSAICVC	Appl% zAAP on CP	Util
SYSLOAVG	Load average	Util
SYSTCTVC	Total CPU time	Util
SYSUCTVC	Uncaptured time	Util
SYSCCTVC	Captured time	Util
SYSCUAVC	Average CPU Util% for zAAPs	No
SYSMUAVC	Average MVS Util% for zAAPs	No
SYSCUIVC	Average CPU Util% for ZIIPs	No
SYSMUIVC	Average MVS Util% for ZIIPs	No

Name	Description of the variable	Report
SYSAHPVC	ZAAP honor priority (YES/NO)	No
SYSIHPVC	ZIIP honor priority (YES/NO)	No
SYSPKCVC	Average number of parked CPs	No
SYSPKAVC	Average number of parked logical processors / threads of type zAAP	No
SYSPKIVC	Average number of parked logical processors / threads of type zIIP	No

SYSRG - Tabular report data table ERBSRGT3

RMF builds ERBSRGT3 when using SYSRG as a report type.

Name	T	Description of the variable	Report
SRGDTLLN	K	Logical line number	-
SRGDTPSN	K	Sequence number	-
SRGNAME	N	Name of resource group or tenant resource group	Yes, Pop-up
SRGTYPE	N	Resource group type	Yes
SRGSYS	N	System name	Yes
SRGSCTRC	N	Associated service class or tenant report class	Popup
SRGSRTYP	N	Definition of minimum and maximum capacity	Yes, Pop-up
SRGSRMIN	N	Service rate (capacity), minimum	Yes, Pop-Up
SRGSRMAX	N	Service rate (capacity), maximum	Yes, Pop-Up
SRGSRACP	N	Service rate (capacity), actual in number of CPs	Yes, Pop-Up
SRGSRAMS	N	Service rate (capacity), actual in MSU	Yes, Pop-Up
SRGSRASU	N	Service rate (capacity), actual in SU/SEC	Yes, Pop-Up
SRGSPINC	N	Include specialty processor consumption	Pop-Up
SRGMMAX	N	Maximum memory limit	Yes
SRGMACT	N	Memory used, actual	Yes
SRGDESCD	N	Resource group description	Pop-Up
SRGSID	N	SMF ID	Pop-Up

SYSRTD - Tabular report data table ERBRTDT3

RMF builds ERBRTDT3 when using SYSRTD as a report type.

Name	T	Description of the variable	Report
RTDDTLLN	K	Logical line number	-
RTDDTPSN	K	Sequence number	-
RTDSYS	N	System identifier	Yes
RTDDAT	N	Data availability indication	Yes
RTDRTWM	N	Wait time / trx in milliseconds	Yes
RTDRTAM	N	Execution time / trx in milliseconds	Yes
RTDRTTM	N	Actual (total) response time / trx in milliseconds	Yes
RTDRTQM	N	Queued time / trx in milliseconds	Pop-Up
RTDRTRM	N	R/S affinity time in milliseconds	Pop-Up
RTDRTIM	N	Ineligible queue time in milliseconds	Pop-Up

Name	T	Description of the variable	Report
RTDRTCM	N	JCL conversion time in milliseconds	Pop-Up
RTDTRAN	N	Ended transactions / second	Yes
RDTSSA	N	Transaction active percentage	Yes
RTDSSR	N	Transaction ready percentage	Yes
RTDSSD	N	Transaction delay percentage	Yes
RTDEXV	N	Execution velocity percentage	Yes
RTDEXD	N	Overall delay percentage	Yes
RTDRTW	N	Wait time / trx in seconds	Util
RDTRTA	N	Execution time / trx in seconds	Util
RTDRTT	N	Actual (total) response time / trx in seconds	Util
RTDRTQ	N	Queued time / trx in seconds	Util
RTDRTR	N	R/S affinity time in seconds	Util
RTDRTI	N	Ineligible queue time in seconds	Util
RTDRTC	N	JCL conversion time in seconds	Util

SYSSUM - Tabular report data table ERBSUMT3

RMF builds ERBSUMT3 when using SYSSUM as a report type.

Name	T	Description of the variable	Report
SUMDTLLN	K	Logical line number	-
SUMDTPSN	K	Sequence number	-
SUMGRP	N	Group name	Yes
SUMTYP	N	Type of WLM group	Yes
SUMIMP	N	Importance of service class period	Yes
SUMVEG	N	Execution velocity goal	Yes
SUMEVA	N	Execution velocity actual	Yes
SUMRTGTM	N	Response time goal in milliseconds	Yes
SUMRTGP	N	Response time goal percentile	Yes
SUMRTATM	N	Response time actual in milliseconds	Yes
SUMRTAP	N	Response time actual percentile	Yes
SUMPFID	N	Performance index	Yes
SUMTRAN	N	Ended transactions / second	Yes
SUMARTWM	N	Wait time in milliseconds	Yes
SUMARTAM	N	Execution time in milliseconds	Yes
SUMARTTM	N	Actual (total) response time in milliseconds	Yes
SUMARTQM	N	Queued time in milliseconds	Util
SUMARTRM	N	R/S affinity time in milliseconds	Util
SUMARTIM	N	Ineligible queue time in milliseconds	Util
SUMARTCM	N	JCL conversion time in milliseconds	Util
SUMGOA	N	Goal type	Util
SUMDUR	N	Duration	Util

Name	T	Description of the variable	Report
SUMRES	N	Name of resource group or tenant resource group	Util
SUMRGTYP	N	Definition of minimum and maximum capacity **SU** service units per second **LS** % of LPAR share **CP** number of CPs x 100 **MS** MSU/h	Util
SUMSMI	N	Service rate (capacity), min.	Util
SUMSMA	N	Service rate (capacity), max.	Util
SUMSRA	N	Service rate (capacity), actual	Util
SUMCRIT	N	Indicator whether Storage Critical or CPU Critical, or both (S, C, or SC)	Util
SUMHONP	N	Indicator whether specialty engine eligible work in this service class will be prevented from being offloaded to CPs for help processing (N)	Util
SUMMLIM	N	Maximum memory limit in GB	Util
SUMEGRP	N	Description of WLM group	Util
SUMRTGT	N	Response time goal in seconds	Util
SUMRTAT	N	Response time actual in seconds	Util
SUMARTW	N	Wait time in seconds	Util
SUMARTA	N	Execution time in seconds	Util
SUMARTT	N	Actual (total) response time in seconds	Util
SUMARTQ	N	Queued time in seconds	Util
SUMARTR	N	R/S affinity time in seconds	Util
SUMARTI	N	Ineligible queue time in seconds	Util
SUMARTC	N	JCL conversion time in seconds	Util
SUMRGSPC	N	Include specialty processor consumption	Util
SUMRCTNT	N	Report class is a tenant report class	Util
SUMMEMUS	N	Memory used, actual	Util

SYSWKM - Tabular report data table ERBWKMT3

RMF builds ERBWKMT3 when using SYSWKM as a report type.

Name	T	Description of the variable	Report
WKMDTLLN	K	Logical line number	-
WKMDTPSN	K	Sequence number	-
WKMJOB	N	Jobname	Yes
WKMASI	N	Address space identification	Yes
WKMSYS	N	System identifier	Yes
WKMSRV	N	Service class name	Yes
WKMSER	N	Service percentage	Yes
WKMPRC	N	Processor using percentage	Yes
WKMVEL	N	Execution velocity percentage	Yes

Name	T	Description of the variable	Report
WKMCAP	N	Capped delay percentage	Yes
WKMQUI	N	Address space quiesced percentage	Yes

USAGE - Tabular report data table ERBJUST3

RMF builds ERBJUST3 when using USAGE as a report type.

Name	Type	Meaning	Report
JUSDTLLN	K	Logical line number	-
JUSDTPSN	K	Sequence number	-
JUSPJOB	N	Jobname	Yes
JUSPASI	N	Address space ID (decimal)	Util
JUSPCLA	N	Job class	Util
JUSPCLAX	N	Extended job class	Yes
JUSPSVCL	N	Service class	Yes
JUSPCLP	N	Service class period	Util
JUSPDP	N	Dispatching priority	Util
JUSPTAT	N	Transaction active time	Util
JUSPTRT	N	Transaction resident time	Util
JUSPTCT	N	Transaction count	Util
JUSPFRT	N	Frames total	Yes
JUSPFRXT	N	Fixed frames total	Yes
JUSPFRXH	N	Fixed frames high	Util
JUSPFRXA	N	Fixed frames between 16M and 2G	Util
JUSPFRXB	N	Fixed frames below 16M	Util
JUSPDCTT	N	Device connect time total since address space creation (in seconds)	Util
JUSPDCTD	N	Device connect time for interval (in seconds)	Yes
JUSPEXCT	N	Total number of EXCP operations since address space creation	Util
JUSPEXCD	N	Number of EXCP operations for interval	Util
JUSPEXCR	N	Number of EXCP operations per second	Yes
JUSPCPUT	N	CPU time total since address space creation (in seconds)	Util
JUSPCPUD	N	CPU time for interval (in seconds)	Yes
JUSPTCBT	N	TCB time total since address space creation (in seconds)	Util
JUSPTCBD	N	TCB time for interval (in seconds)	Yes
JUSPQREQ	N	GQSCAN requests	Yes
JUSPQSPR	N	GQSCAN specific requests	Util
JUSPQRES	N	Average number of GQSCAN resources	Yes
JUSPQRSD	N	GQSCAN resource count standard deviation	Util
JUSPQTIM	N	Average GQSCAN request time	Yes
JUSPQTSD	N	GQSCAN request time standard deviation	Util

WFEX - Tabular report data table ERBWFXT3

RMF builds ERBWFXT3 when using WFEX as a report type.

Name	T	Description of the variable	Report
WFXDTLLN	K	Logical line number	-
WFXDTPSN	K	Sequence number	-
WFXATTR	N	Attribute	Util
WFXNAME	N	Name	Yes
WFXREASN	N	Reason	Yes
WFXDELAY	N	Delay	Yes
WFXPCAUS	N	Possible cause	Yes

XCF - Tabular report data table ERBXCFT3

RMF builds ERBXCFT3 when using XCF as a report type.

Name	T	Description of the variable	Report
XCFDTLLN	K	Logical line number	-
XCFDTPSN	K	Sequence number	-
XCFPJOB	N	Jobname	Yes
XCFPCLA	N	Class (A, B, O, S, or T)	Yes
XCFPDMN	N	Domain number; no longer used	Yes
XCFPPGN	N	Performance group number; no longer used	Yes
XCFPSVCL	N	Service class name	Yes
XCFPODEL	N	Overall delay	Yes
XCF1SDEL	N	Delay percentage (Path 1)	Yes
XCF1PATH	N	Path 1	Yes
XCF2SDEL	N	Delay percentage (Path 2)	Yes
XCF2PATH	N	Path 2	Yes
XCF3SDEL	N	Delay percentage (Path 3)	Yes
XCF3PATH	N	Path 3	Yes
XCF4SDEL	N	Delay percentage (Path 4)	Yes
XCF4PATH	N	Path 4	Yes

ZFSFS - Tabular report data table ERBZFFT3

RMF builds ERBZFFT3 when using ZFSFS as a report type.

Name	Type	Meaning	Report
ZFFDTLLN	K	Logical line number	–
ZFFDTPSN	K	Sequence number	–
ZFFPFILE	N	File system name line or file system detailed data line containing the 9 values following the file system name	Yes
ZFFPFSN	N	File system name	Pop-up
ZFFPSYSC	N	Connected system	Pop-up
ZFFPSYSO	N	Owning system	Pop-up
ZFFPMODE	N	File system mode	Pop-up
ZFFPSIZE	N	Maximum size	Util
ZFFPUSGP	N	Currently used space percentage	Util

Name	Type	Meaning	Report
ZFFPAPIR	N	I/O rate (read + write) by applications	Util
ZFFPAPRT	N	Response time (read + write) by applications	Util
ZFFPAPRP	N	Percentage of read operations in I/O rate	Util
ZFFPAPXR	N	XCF rate (read + write)	Util
ZFFPFSMP	N	Mount point	Pop-up
ZFFPFSVN	N	Vnodes	Pop-up
ZFFPFSVU	N	USS held vnodes	Pop-up
ZFFPOBJO	N	Open objects	Pop-up
ZFFPOBJT	N	Tokens	Pop-up
ZFFPFSUC	N	User cache 4K pages	Pop-up
ZFFPFSMC	N	Metadata cache 8K pages	Pop-up
ZFFPAPRR	N	Application read rate	Pop-up
ZFFPARRT	N	Application read response time	Pop-up
ZFFPXFRR	N	XCF read rate	Pop-up
ZFFPXRRT	N	XCF read response time	Pop-up
ZFFPIORR	N	Aggregate read rate	Pop-up
ZFFPAPWR	N	Application write rate	Pop-up
ZFFPAWRT	N	Application write response time	Pop-up
ZFFPXFWR	N	XCF write rate	Pop-up
ZFFPXWRT	N	XCF write response time	Pop-up
ZFFPIOWR	N	Aggregate write rate	Pop-up
ZFFPESPC	N	ENOSPC errors	Pop-up
ZFFPEDIO	N	Disk I/O errors	Pop-up
ZFFPEXCF	N	XCF communication failures	Pop-up
ZFFPOPCA	N	Cancelled operations	Pop-up

ZFSKN - Tabular report data table ERBZFKT3

RMF builds ERBZFKT3 when using ZFSKN as a report type.

Name	T	Description of the variable	Report
ZFKDTLLN	K	Logical line number	–
ZFKDTPSN	K	Sequence number	–
ZFKPSYS	N	System name	Yes
ZFKPRQRL	N	Request rate local	Yes
ZFKPRQRR	N	Request rate remote	Yes
ZFKPXFRL	N	XCF rate local	Yes
ZFKPXFRR	N	XCF rate remote	Yes
ZFKPRPTL	N	Response time local	Yes
ZFKPRPTR	N	Response time remote	Yes

ZFSOVW - Tabular report data table ERBZFOT3

RMF builds ERBZFOT3 when using ZFSOVW as a report type.

Name	Type	Meaning	Report
ZFODTLLN	K	Logical line number	–
ZFODTPSN	K	Sequence number	–
ZFOPSYS	N	System name	Yes
ZFOPIORP	N	Avg response time I/O%	Yes
ZFOPLORP	N	Avg response time Lock%	Yes
ZFOPSLRP	N	Avg response time Sleep%	Yes
ZFOPUCRT	N	User cache request rate	Yes
ZFOPUCHP	N	User cache hit%	Yes
ZFOPVCRT	N	Vnode cache request rate	Yes
ZFOPVCHP	N	Vnode cache hit%	Yes
ZFOPMCRT	N	Metadata cache request rate	Yes
ZFOPMCHP	N	Metadata cache hit%	Yes
ZFOPITYn	N	IO summary: type of I/O request (where n is 1, 2, 3)	Pop-up
ZFOPICTn	N	IO summary: total number of I/O requests (where n is 1, 2, 3)	Pop-up
ZFOPIWTn	N	IO summary: number of I/O requests waiting (where n is 1, 2, 3)	Pop-up
ZFOPICAn	N	IO summary: number of I/O requests cancelled (where n is 1, 2, 3)	Pop-up
ZFOPIMGn	N	IO summary: number of I/O requests merged (where n is 1, 2, 3)	Pop-up
ZFOPUCSZ	N	User cache size	Pop-up
ZFOPUCSF	N	User cache storage fixed	Pop-up
ZFOPUCTP	N	User cache number of total pages	Pop-up
ZFOPUCFP	N	User cache number of free pages	Pop-up
ZFOPUCSG	N	User cache number of segments	Pop-up
ZFOPUCRR	N	User cache read rate	Pop-up
ZFOPUCRH	N	User cache read hit percentage	Pop-up
ZFOPUCRD	N	User cache read delay percentage	Pop-up
ZFOPUCAR	N	User cache asynchronous read rate	Pop-up
ZFOPUCWR	N	User cache write rate	Pop-up
ZFOPUCWH	N	User cache write hit percentage	Pop-up
ZFOPUCWD	N	User cache write delay percentage	Pop-up
ZFOPUCSW	N	User cache scheduled writes rate	Pop-up
ZFOPUCRP	N	User cache read%	Pop-up
ZFOPUCDP	N	User cache dly%	Pop-up
ZFOPUCRW	N	User cache number of page reclaim writes	Pop-up
ZFOPUCFS	N	User cache number of file synchronization requests	Pop-up
ZFOPVCSZ	N	Vnode cache size	Pop-up
ZFOPVCAL	N	Vnode cache number of currently allocated vnodes	Pop-up
ZFOPVCSN	N	Vnode cache size of a vnode data structure	Pop-up
ZFOPVCEX	N	Vnode cache number of extended vnodes	Pop-up

Name	Type	Meaning	Report
ZFOPVCSE	N	Vnode cache size of an extended vnode	Pop-up
ZFOPVCOP	N	Vnode cache number of open vnodes	Pop-up
ZFOPVCHE	N	Vnode cache number of held vnodes	Pop-up
ZFOPVCRQ	N	Vnode cache total number of requests	Pop-up
ZFOPVCCR	N	Vnode cache number of requests to allocate (create) vnodes	Pop-up
ZFOPVCDL	N	Vnode cache number of requests to delete vnodes	Pop-up
ZFOPMCSZ	N	Metadata cache size	Pop-up
ZFOPMCSF	N	Metadata cache storage fixed	Pop-up
ZFOPMCBU	N	Metadata cache number of buffers	Pop-up
ZFOPMCRQ	N	Metadata cache total number of requests	Pop-up
ZFOPMCUD	N	Metadata cache number of updates	Pop-up
ZFOPMCPW	N	Metadata cache number of partial writes	Pop-up

Graphic report parameter table ERBPTGS3

The graphic report parameter table defines the layout of graphic reports for panel display and hardcopy printing. The first part describes general information about the graphic report. The second part describes information about labels per bar. The third part describes the column layout.

The format for general information is:

Name	T	Description of the variable	Example
PTGREPNA	K	Report name (must be specified)	DEV HSM JES
PTGRHELP	N	Name for help panel – See name convention for HELP panels	
PTGRMINY	N	Length of Y-scale, if there is no bar exceeding this length. **1** for average number of user's time, **100** for percentage values	1 100
PTGRAXTI	N	Title of the axis • Percentage of Each User's Time • Percentage of The User's Time • Average Number of Active Users	1 100

Name	T	Description of the variable	Example
PTGRSERU	N	Selection rule for bars: **0 :** One bar corresponds to one line **1 :** One bar corresponds to one line with the sequence number 1 **2 :** One bar corresponds to the summary of logical lines **3 :** Two bar-types result from all logical lines of a logical block • Bar type 1 corresponds to sequence number 1 • Bars of bar type 2 correspond to each line of the logical block	0 1 2 3 DELAY DEV, HSM, JES DEVR, ENQR STORR
PTGRBRNM	N	Number of bar types ‘1’ and ‘2’, represented by the character before the last character in the following variables.	1 2

The format for labels per bar is:

Variable Name	T	Variable Description	Example
PTGRLB10	N	Number of labels per bar for bar type 1	1 2
PTGRCL1 PTGRCL2	N N	ISPF COLUMN data-table variables containing the labels for bar-type 1.	
PTGRAP1 PTGRAP2	N N	Alpha part of the labels, which will be composed by this part and the last 4 digits of the data value.	‘DMN’, ‘PG’ in DELAY
PTGRLB20	N	Number of labels per bar for bar type 2	1 2
PTGRCL3 PTGRCL4 PTGRAP3 PTGRAP4	N N N N	(corresponding to PTGRCL1) (corresponding to PTGRCL2) (corresponding to PTGRAP1) (corresponding to PTGRAP2)	

The format for columns is:

Variable Name	T	Variable Description	Example
PTGRCPNM PTGRTV1 PTGRLD1 PTGRAL1 PTGRDC1	N N N N N	Number of data columns to be selected for the bar types. = number of color-pattern-text combin. (0, 1, 2, ... 9, represented by the last character of the variable. ISPF Column Table variable. This variable contains a specific data value of the tabular report after a TBGET to a row of the Data Column Table. (Must be specified) Legend ID, to specify a particular color-pattern-text combination of the Color-Pattern Table. The ID specifies the legend (color, pattern and subheader) for this data value. Transformation ID 0 : don't divide 1 : divide by 10 2 : divide by 100 bartype col ; If '0', the data value 0 : reports value in both bar types 1 : reports value in first bar type 2 : reports value in second bar type	0 1 ... 9 1 2 ... see color-pattern option table 0 2 0 1 2
PTGRTV2 PTGRLD2 PTGRAL2 PTGRDC2	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV3 PTGRLD3 PTGRAL3 PTGRDC3	N N N N	(corresponding to ptgrtv1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV4 PTGRLD4 PTGRAL4 PTGRDC4	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV5 PTGRLD5 PTGRAL5 PTGRDC5	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV6 PTGRLD6 PTGRAL6 PTGRDC6	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV7 PTGRLD7 PTGRAL7 PTGRDC7	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV8 PTGRLD8 PTGRAL8 PTGRDC8	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	

Variable Name	T	Variable Description	Example
PTGRTV9 PTGRLD9 PTGRAL9 PTGRDC9	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	
PTGRTV10 PTGRLD10 PTGRAL10 PTGRDC10	N N N N	(corresponding to PTGRTV1) (corresponding to PTGRLD1) (corresponding to PTGRAL1) (corresponding to PTGRDC1)	

RMF Phase driver table ERBPHDS3

The phase driver table has rows for each command and selection.

Variable Name	Type	Variable Description
PHDREPNA	K	Name of the command or the long form of the report selection.
PHDREPSE	N	Selection string to be created. This string will be passed to the primary option panel to perform the command function.
PHDRPH1	N	Function to be performed for Phase 1. The string if not null, will be selected.
PHDRPH2	N	Function to be performed for Phase 2. The string if not null, will be selected.
PHDRPH3	N	Function to be performed for Phase 3. The string if not null, will be selected.
PHDRPH4	N	Function to be performed for Phase 4. The string if not null, will be selected.
PHDRET1	N	Return code passed from Phase 1. The Phase 2 and Phase 3 are executed only if the return code from this Phase is zero.
PHDRET2	N	Return code passed from Phase 2. The Phase 3 is executed only if the return code from this Phase is zero.
PHDRET3	N	Return code passed from Phase 3.
PHDRET4	N	Return code passed from Phase 4.
PHDRTAB1	N	Name of the ISPF table created by Phase 1. This table is input for Phase 2.
PHDRTAB2	N	Name of the ISPF table created by Phase 2. This table is input to Phase 3.

The following table lists the report commands, selections, and the variables used for each phase (1,2,3,4). Phases 2 and 4 are null.

PHDREPNA	PHDREPSE	PHDRPH1	PHDRPH3	PHDRTAB1
CACHDET	S.9	PGM(ERB3RPH1) PARM(CACHDET)	PGM(ERB3RDSP)	ERBCADT3
CACHSUM	S.8	PGM(ERB3RPH1) PARM(CACHSUM)	PGM(ERB3RDSP)	ERBCAST3
CFACT	S.7	PGM(ERB3RPH1) PARM(CFACT)	PGM(ERB3RDSP)	ERBCFAT3
CFOVER	S.5	PGM(ERB3RPH1) PARM(CFOVER)	PGM(ERB3RDSP)	ERBCFOT3
CFSYS	S.6	PGM(ERB3RPH1) PARM(CFSYS)	PGM(ERB3RDSP)	ERBCFST3
CHANNEL	3.12	PGM(ERB3RPH1) PARM(CHANNEL)	PGM(ERB3RDSP)	ERBCHAT3
CPC	1.3	PGM(ERB3RPH1) PARM(CPC)	PGM(ERB3RDSP)	ERBCPCT3
CRYACC	S.17	PGM(ERB3RPH1) PARM(CRYACC)	PGM(ERB3RDSP)	ERBCRYT3
CRYOVW	S.16	PGM(ERB3RPH1) PARM(CRYOVW)	PGM(ERB3RDSP)	ERBCRYT3
CRYPKC	S.18	PGM(ERB3RPH1) PARM(CRYPKC)	PGM(ERB3RDSP)	ERBCRYT3
DELAY	1.4	PGM(ERB3RPH1) PARM(DELAY)	PGM(ERB3RDSP)	ERBJDET3
DEV	3.2	PGM(ERB3RPH1) PARM(DEV)	PGM(ERB3RDSP)	ERBDEVT3

PHDREPNA	PHDREPSE	PHDRPH1	PHDRPH3	PHDRTAB1
DEVR	3.3	PGM(ERB3RPH1) PARM(DEVR)	PGM(ERB3RDSP)	ERBDVRT3
DSINDEX	S.D	PGM(ERB3RHP1) PARM(DSINDEX)	PGM(ERB3RDSP)	ERBDSIT3
DSND	3.3A	PGM(ERB3RPH1) PARM(DSND)	PGM(ERB3RDSP)	ERBDNDT3
DSNJ	2.1A	PGM(ERB3RPH1) PARM(DSNJ)	PGM(ERB3RDSP)	ERBDNJT3
DSNV	3.3B	PGM(ERB3RPH1) PARM(DSNV)	PGM(ERB3RDSP)	ERBDNVT3
EADM	3.15	PGM(ERB3RPH1) PARM(EADM)	PGM(ERB3RDSP)	ERBSCMT3
ENCLAVE	1.6	PGM(ERB3RPH1) PARM(ENCLAVE)	PGM(ERB3RDSP)	ERBENCT3
ENQ	3.4	PGM(ERB3RPH1) PARM(ENQ)	PGM(ERB3RDSP)	ERBENQT3
ENQR	3.5	PGM(ERB3RPH1) PARM(ENQR)	PGM(ERB3RDSP)	ERBEQRT3
GROUP	1.5	PGM(ERB3RPH1) PARM(GROUP)	PGM(ERB3RDSP)	ERBGRTT3
HSM	4.1	PGM(ERB3RPH1) PARM(HSM)	PGM(ERB3RDSP)	ERBHSMT3
IOQ	3.13	PGM(ERB3RPH1) PARM(IOQ)	PGM(ERB3RDSP)	ERBIOQT3
JES	4.2	PGM(ERB3RPH1) PARM(JES)	PGM(ERB3RDSP)	ERBJEST3
JOB	2.5	PGM(ERB3RPH1) PARM(JOB)	PGM(ERB3RDSP)	ERBJDJT3
OPD	1.7	PGM(ERB3RPH1) PARM(OPD)	PGM(ERB3RDSP)	ERBOPDT3
PCIE	3.14	PGM(ERB3RPH1) PARM(PCIE)	PGM(ERB3RDSP)	ERBPCIT3
PROC	3.1	PGM(ERB3RPH1) PARM(PROC)	PGM(ERB3RDSP)	ERBPRCT3
RLSDS	S.11	PGM(ERB3RPH1) PARM(RLSDS)	PGM(ERB3RDSP)	ERBVRDT3
RLSLRU	S.12	PGM(ERB3RPH1) PARM(RLSLRU)	PGM(ERB3RDSP)	ERBVRLT3
RLSSC	S.10	PGM(ERB3RPH1) PARM(RLSSC)	PGM(ERB3RDSP)	ERBVRST3
STOR	3.6	PGM(ERB3RPH1) PARM(STOR)	PGM(ERB3RDSP)	ERBSTRT3
STORC	3.10	PGM(ERB3RPH1) PARM(STORC)	PGM(ERB3RDSP)	ERBCSUT3
STORCR	3.11	PGM(ERB3RPH1) PARM(STORCR)	PGM(ERB3RDSP)	ERBCRST3
STORF	3.7	PGM(ERB3RPH1) PARM(STORF)	PGM(ERB3RDSP)	ERBSTFT3
STORR	3.8	PGM(ERB3RPH1) PARM(STORR)	PGM(ERB3RDSP)	ERBSRRT3
STORS	3.9	PGM(ERB3RPH1) PARM(STORS)	PGM(ERB3RDSP)	ERBSRST3
SYSENQ	S.4	PGM(ERB3RPH1) PARM(SYSENQ)	PGM(ERB3RDSP)	ERBEQST3
SYSINFO	1.2	PGM(ERB3RPH1) PARM(SYSINFO)	PGM(ERB3RDSP)	ERBSYST3
SYSRG	S.1A	PGM(ERB3RPH1) PARM(SYSRG)	PGM(ERB3RDSP)	ERBSRGT3
SYSRTD	S.2	PGM(ERB3RPH1) PARM(SYSRTD)	PGM(ERB3RDSP)	ERBRTDT3
SYSSUM	S.1	PGM(ERB3RPH1) PARM(SYSSUM)	PGM(ERB3RDSP)	ERBSUMT3
SYSWKM	S.3	PGM(ERB3RPH1) PARM(SYSWKM)	PGM(ERB3RDSP)	ERBWKMT3
USAGE	1.4A	PGM(ERB3RPH1) PARM(USAGE)	PGM(ERB3RDSP)	ERBJUST3
WFEX	1.1	PGM(ERB3RPH1) PARM(WFEX)	PGM(ERB3RDSP)	ERBWFXT3
XCF	4.3	PGM(ERB3RPH1) PARM(XCF)	PGM(ERB3RDSP)	ERBXCFT3
ZFSFS	S.14	PGM(ERB3RPH1) PARM(ZFSFS)	PGM(ERB3RDSP)	ERBZFFT3
ZFSKN	S.15	PGM(ERB3RPH1) PARM(ZFSKN)	PGM(ERB3RDSP)	ERBZFKT3
ZFSOVW	S.13	PGM(ERB3RPH1) PARM(ZFSOVW)	PGM(ERB3RDSP)	ERBZFOT3

Appendix A. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

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

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

*** indicates an optional syntax element that is repeatable**
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area.

If you hear the lines `3* , 3 HOST, 3 STATE`, you know that you can include `HOST, STATE`, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write `HOST STATE`, but you cannot write `HOST HOST`.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line `6.1+ data area`, you must include at least one data area. If you hear the lines `2+, 2 HOST,` and `2 STATE`, you know that you must include `HOST, STATE,` or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

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 the Knowledge Centers. 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

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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 DFSMS, 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.

Exploitation of the Flash Express feature

IBM intends to provide exploitation of the Flash Express® feature on IBM zEnterprise® EC12 (zEC12) and zBC12 servers with CFLEVEL 19 for certain coupling facility list structures in the first half of 2014. This new function is designed to allow list structure data to migrate to Flash Express memory as needed, when the consumers of data do not keep pace with its creators for some reason, and migrate it back to real memory to be processed. When your installation uses WebSphere® MQ for z/OS Version 7 (5655-R36), this new capability is expected to provide significant buffering against enterprise messaging workload spikes and provide support for storing large amounts of data in shared queue structures, potentially allowing several hours' data to be stored without causing interruptions in processing. In addition, z/OS V2R1 Resource Measurement Facility (RMF) is planned to provide measurement data and reporting capabilities for Flash Express when it is used with coupling facilities. Information about externals and interfaces that are related to this planned capability are being made available in z/OS V2R1 for early planning and development purposes only.

Programming Interface Information

This book documents intended Programming Interfaces that help customers to write their own RMF exit routines and to call RMF functions from their own applications.

Trademarks

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 RMF and MVS terms are also defined.

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

APPC/MVS
Advanced program-to-program communication

ASCH address space
APPC transaction scheduler address space

AS
Address space

address space
That part of MVS main storage that is allocated to a job.

auxiliary storage (AUX)
All addressable storage, other than main storage, that can be accessed by means of an I/O channel; for example storage on direct access devices.

background session
In RMF, a monitor session that is started and controlled from the operator console. Contrast with *interactive session*

balanced systems
To avoid bottlenecks, the system resources (CP, I/O, storage) need to be balanced.

basic mode
A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

bottleneck
A system resource that is unable to process work at the rate it comes in, thus creating a queue.

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

CIM provider
A CIM provider is the link between the CIM server and the system interfaces. It allows the CIM server to access and manage the resources. Each CIM provider exposes the resources it represents in a standard way, using a small number of classes from the CIM schema or derived from the CIM schema. RMF monitoring providers are CIM providers implemented by RMF.

contention
Two or more incompatible requests for the same resource. For example, contention occurs if a user requests a resource and specifies exclusive use, and another user requests the same resource, but specifies shared use.

coupling facility
See *Cross-system Extended Services/Coupling Facility*.

CP
Central processor

criteria
Performance criteria set in the WFEX report options. You can set criteria for all report classes (PROC, SYSTEM, TSO, and so on).

CPU speed
Measurement of how much work your CPU can do in a certain amount of time.

cross-system coupling facility (XCF)
A component of MVS that provides functions to support cooperation between authorized programs running within a *sysplex*.

Cross-system Extended Services/Coupling Facility (XES/CF)
Provides services for MVS systems in a sysplex to share data on a coupling facility (CF).

CS
Central storage

Customer Information Control System (CICS)
An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining data bases.

cycle
In RMF, the time at the end of which one sample is taken. Varies between 50 ms and 9999 ms. See also *sample*.

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.

exception reporting
In RMF, the reporting of performance measurements that do not meet user-defined criteria. Shows potential performance problems explicitly, thus avoiding the need for constant monitoring.

generalized trace facility (GTF)
A service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

GO mode
In RMF, the Monitor III mode in which the screen is updated with the interval you specified in your session options. The terminal cannot be used for anything else when it is in GO mode. See also *mode*.

graphic mode
In RMF Monitor III, the mode which presents the performance data from the system in graphic format using the GDDM product. Contrast with *tabular mode*.

GTF
generalized trace facility

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.

mode
Monitor III can run in various modes: GO mode (see *GO mode*) and STOP mode, which is the default mode. See also *graphic mode* and *tabular mode*.

MPL
Multiprogramming level

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.

RMF monitoring provider
see CIM provider

sample
Once in every cycle, the number of jobs waiting for a resource, and what job is using the resource at that moment, are gathered for all resources of a system by Monitor III. These numbers constitute one sample.

SCP
System control program

seek
The DASD arm movement to a cylinder. A seek can range from the minimum to the maximum seek time of a device. In addition, some I/O operations involve multiple imbedded seeks where the total seek time can be more than the maximum device seek time.

service class
In Workload Manager, a subdivision of a *workload*. Performance goals and capacity boundaries are assigned to service classes.

service level agreement (SLA)
A written agreement of the information systems (I/S) service to be provided to the users of a computing installation.

Service Level Reporter (SLR)
An IBM licensed program that provides the user with a coordinated set of tools and techniques and consistent information to help manage the data processing installation. For example, SLR extracts information from SMF, IMS, and CICS logs, formats selected information into tabular or graphic reports, and gives assistance in maintaining database tables.

service rate
In the system resources manager, a measure of the rate at which system resources (services) are provided to individual jobs. It is used by the installation to specify performance objectives, and used by the workload manager to track the progress of individual jobs. Service is a linear combination of processing unit, I/O, and main storage measures that can be adjusted by the installation.

shared page groups
An address space can decide to share its storage with other address spaces using a function of RSM. As soon as other address spaces use these storage areas, they can no longer be tied to only one address space. These storage areas then reside as *shared page groups* in the system. The pages of shared page groups can reside in central, expanded, or auxiliary storage.

SLA
service level agreement

SLIP
serviceability level indication processing

SLR
Service Level Reporter

SMF
System management facility

SMF buffer
A wrap-around buffer area in storage, to which RMF data gatherers write performance data, and from which the Postprocessor extracts data for reports.

speed
See *workflow*

SRB
Service request block

SRM
System resource manager

SSCH
Start subchannel

system control program (SCP)
Programming that is fundamental to the operation of the system. SCPs include MVS, VM, and VSE operating systems and any other programming that is used to operate and maintain the system. Synonymous with *operating system*.

sysplex
A complex consisting of a number of coupled MVS systems.

tabular mode
In RMF, the mode in which Monitor III displays performance data in the form of lists. Contrast with *graphic mode*.

TCB
Task control block

threshold
The exception criteria defined on the report options screen.

throughput
A measure of the amount of work performed by a computer system over a period of time, for example, number of jobs per day.

TPNS
Teleprocessing network simulator

TSO
Time Sharing Option, see *Time Sharing Option/Extensions*

Time Sharing Option Extensions (TSO/E)
In MVS, a time-sharing system accessed from a terminal that allows user access to MVS system services and interactive facilities.

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

workflow
The workflow of an address space represents how a job uses system resources and the speed at which the job moves through the system in relation to the maximum average speed at which the job could move through the system.

The workflow of resources indicates how efficiently users are being served.

workload
A logical group of work to be tracked, managed, and reported as a unit. Also, a logical group of service classes.

WLM
Workload Manager

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

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Product Number: 5650-ZOS

GC27-4937-40