IBM OMEGAMON for IMS on z/OS
5.5.0

Historical Component (EPILOG) Reference

Note

Before using this information and the product it supports, read the information in "Notices" on page 155.

September 2024 Edition

This edition applies to version 5, release 5, modification 0, of IBM OMEGAMON for IMS on z/OS (product number 5698-T02) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Introduction to the EPILOG historical component

The EPILOG historical component of OMEGAMON for IMS provides IMS performance data for days, weeks, or months. Data samples are taken and stored in an EPILOG datastore (EDS). The EDS is a historical database of IMS system performance data, which you can review interactively or through batch reports.

In TSO, you can view information that is contained in the EDS. You can analyze performance degradation for any time period and be better informed about how to improve future performance. EPILOG shows statistics historically, and provides information over periods of time. Using EPILOG you can review the causes of poor performance after the fact, which extends the capabilities of real time monitors.

EPILOG also includes standard batch reports. You use pre-established sets of commands to get reports on the capacity, performance, or service level of a data center. You can modify standard reports for specific situations or you can create new reports.

The basis of the EPILOG historical component is the IBM IMS tuning philosophy, which focuses on service-level analysis, bottleneck analysis, and resource analysis. Therefore, EPILOG can assist you with the following tasks:

- Achieving service-level commitments through analysis of transaction response times.
- Recognizing poor response time or throughput and providing bottleneck statistics to identify major bottlenecks.
- Review resources that are associated with bottlenecks to pinpoint the precise component that needs improvement.
- Evaluating how effective recent changes are in correcting either hardware or software bottlenecks.

EPILOG product overview

The EPILOG historical component consists of three operating components: the *collector*, the *reporter*, and the *maintenance utilities*. You use the collector and the reporter to access and report systems data. You use the utilities to periodically maintain the EPILOG datastore (EDS).

EPILOG collector
The subcomponent that gathers IMS response times, bottleneck, and other operational information and writes it into the EDS.

EPILOG reporter
The subcomponent that reads the EDS data and reports either interactively (by using TSO) or through batch processing.

EPILOG maintenance utilities
The maintenance facilities that you use to periodically maintain the EPILOG data store.

SMF data set
The optional z/OS® system data set that stores system performance data.

EPILOG datastore
The VSAM key-sequenced data set, which is the KSDS data set, that organizes and stores collected data over time.

Tip: For more comprehensive reporting on the performance of your IMS systems, IMS Performance Analyzer offers a suite of reports to help you manage the performance and resource utilization of your IMS systems. IMS Performance Analyzer processes IMS log, IMS monitor, IMS Connect event data written to IMS Connect Extensions journals, and IBM OMEGAMON® for IMS Application Trace Facility (ATF) data to provide comprehensive reports for use by IMS specialists to tune their IMS systems, and managers to verify service levels and predict trends. For more information, refer to the *IMS Performance Analyzer User's Guide*.

The EPILOG collector

The OMEGAMON for IMS standard installation procedure sets the EPILOG collector to start automatically when the OMEGAMON for IMS address space starts.

You can also start the EPILOG collector separately with the interface procedure KEICOL*mp*, which issues the following interface command:

```
START EPILOG,OPTION=KEIOPTmp
```

Alternatively, you can issue the interface **START EPILOG** command.

For a specific IMS system, only one collector can be active at one time. The collector gathers several types of performance-related information:

- Service levels (response times by transaction group)
- Bottleneck data (by transaction group)
- Resource usage data
- Node data

By default, at the end of each resource measurement facility (RMF) collection interval (typically 15 minutes to 1 hour), the collector writes the data that it collected during the interval to SMF and the EPILOG datastore (EDS). Optionally, you can specify another collection interval. For more information, see "Options for the EPILOG collector" on page 13.

EPILOG does not duplicate information that is collected by RMF. However, if you do not synchronize on the RMF interval, certain device statistics are not available.

You can specify that EPILOG write all the collected data to SMF. Turn on this option so you do not have to make backup copies of the EPILOG datastore. If required, you can use EDS utilities to re-create the EDS from SMF records.

The EPILOG reporter

You can use the EPILOG reporter for interactive inquiry or to generate batch reports. You can issue commands to request data for specific time periods, combine and average collection intervals, analyze periods with exceptional conditions, and customize report formats.

For example, you can use the EPILOG reporter to complete the following tasks:

- Find all periods in which the average IMS response time exceeded 3 seconds.
- Graphically display IMS response time for a day in the past (for instance, yesterday) and highlight the major cause of degradation corresponding to each collection period.
- Review resource usage statistics that are associated with each degradation reason for that period.
- Find up to 10 transactions that have more than 15 seconds of elapsed processing time.
- Display the daily average response time for the INQUIRY transaction during the last month and show the leading cause of degradation for each day.
- Identify transactions that have an average input queue wait of over 3 seconds.
- Report periods where transaction processing time was delayed because of long running batch message processing (BMP) applications.
- Show average transaction response time during selected time periods.
- Display the average bottleneck profile for Class 1 transactions during the past month.
- Display average response time and the corresponding bottleneck profile for Class 1 transactions before and after a real storage upgrade was implemented.

The EPILOG reporter, operating as a batch job, produces reports equivalent to those reports that run under TSO. The reporter includes a set of standard reports with the information that is required to improve user service. You can produce any reports that are too long to display on a terminal in their entirety in batch mode. You can also customize regular reports for review at the end of each shift, week, or month.

The EPILOG data store (EDS)

The EPILOG data store can contain data that is collected from different versions of IMS.

However, only one EPILOG collector can write data to an EDS at any time. For example, if two different EPILOG collectors are each monitoring a different IMS, the collectors cannot both write to the same EDS. Also, if your installation is running OMEGAMON on z/OS, you must allocate a new EDS to store the OMEGAMON for IMS data. The EDS cannot be used as a common database for z/OS and IMS data.

The EPILOG data store (EDS) maintenance utilities

IBM provides the utilities that maintain EDS data. These utilities can run statistical analysis of disk space usage, eliminate data that is no longer useful, and reconstruct EDS data from SMF records.

EPILOG performance reporting

You can use EPILOG for basic performance reporting tasks. All EPILOG displays are available interactively (under TSO) and in batch. Under TSO, the reporter takes advantage of the "fullscreen" capabilities of an IBM 3270 family terminals. Scrolling and highlighting of important information, and extended color support, is available with such 3270 terminals.

You can request EPILOG information panels by issuing simple yet powerful commands, for example:

```
DISPLAY SUMMARY GROUP(3) YESTERDAY REPORTIF(R0TIME(>1S)) -
        STARTTIME(1000) ENDTIME(1200)
```

These commands generate a panel with a line for each RMF interval yesterday between 10 AM and 12 noon in which transaction group 3's response time 0 is greater than 1 second. In addition, the dominant cause of bottleneck is displayed, along with its impact on the total response time. The response would look like the display that is shown in Figure 1 on page 3.

```
TRANSACTION GROUP =   3
=============================================================================
 PERIOD: 10:00 to 12:00 on 08/29/08
=============================================================================
. COMPETING TRANSACTIONS
0==============================================================================
 DATE__START_END___MAIN_REASON(*)_TIME(-)|0S___0.2____0.4____0.6____0.8_____10
 02/08 10:00 10:15  USING CPU/APPL 2.02 S|***---->      .      .      .      .
       10:00 10:15  CPU WAIT       1.36 S|**---> .      .      .      .      .
       10:15 10:30  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .
       10:30 10:45  DB PMFG021     7.06 S|*****************-------->  .      .
       10:45 11:00  DB PMFG021     9.85 S|**************************-------->.
       11:00 11:15  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .
       11:15 11:30  DB PMFG021     5.38 S|*************------> .      .      .
       11:30 11:45  DB PACCT03     1.73 S|***--->.      .      .      .      .
       11:45 12:00  DB PMFG021     2.94 S|***------->   .      .      .      .
 Average (Response Time 0)
```

Figure 1. Summary display

There are five basic EPILOG displays:

- Group summary (GSUM) display. See Figure 20 on page 32 in "Transaction group summary (GSUM keyword)" on page 31.
- Summary display. See Figure 1 on page 3.
- Detailed bottleneck display, which subdivides an average transaction processing time into its component execution states. See Figure 13 on page 25 in "Detailed workload (DETAIL keyword)" on page 24.
- Resource displays, which show traditional resource usage statistics. See Chapter 11, "Reference: Sample resource panels in EPILOG displays," on page 101.

- IMS CIOP statistics
  - IMS combined pool statistics (SPAP, MAIN, DMBW, PSBW)
  - IMS processor activity
  - IMS data set statistics
  - IMS device statistics
  - IMS DMB pool statistics
  - IMS FRE pool statistics
  - General IMS and z/OS system information
  - IMS latch statistics
  - IMS logging statistics
  - IMS message format buffer pool statistics
  - IMS ISAM/OSAM buffer pool statistics
  - IMS paging and storage information
  - IMS PI enqueue pool statistics
  - IMS PSB pool statistics
  - IMS QBUF statistics
  - IMS SAP pool statistics
  - IMS scheduling statistics
  - IMS SRM statistics
  - Display all resource panels
  - IMS sequential buffering statistics
  - IMS receive any pool statistics
- Collection options and group definitions display. See Figure 21 on page 34 in "Collection options (COPT keyword)" on page 34).

Note: If you are using EPILOG in a Database Control (DBCTL) environment, your displays might look different from the displays in the following topics.

Approach to performance tuning with EPILOG

You can use EPILOG to analyze performance problems after they occur. The IBM tuning approach consists of three basic steps: service-level analysis, bottleneck analysis, and resource analysis.

Service-level analysis
: To improve performance, you must first establish service goals. The data center and its users must agree on what is acceptable service by defining average transaction response times. The first step in IBM's tuning approach identifies the time periods when response time exceeds these commitments.

Bottleneck analysis
: Service-level analysis identifies periods of slow response without pinpointing the causes. Bottleneck analysis determines bottlenecks responsible for poor response time. For example, bottleneck analysis can show that when response time is 2 seconds, certain transactions are spending 50 percent of their time (1 second) awaiting a message processing program (MPP) to be scheduled to service the transactions.

Resource analysis
: After you discover periods of poor response time and their causes, you analyze associated resource usage information to pinpoint why these transactions are waiting. EPILOG provides a set of resource displays.

You can use the system navigator to complete logical tuning process. The system navigator is a facility that finds periods of poor response time and the bottlenecks that might be at fault. For more information, see Chapter 6, "EPILOG system navigator," on page 65.

Bottleneck analysis

The EPILOG collector uses a type of statistical analysis that is called bottleneck analysis. The collector is the part of EPILOG that gathers data about where and how long a workload spends time. The various states in which a transaction can spend time are called execution states.

The following are the execution states:

- Using CPU
- Scheduling waits
- Database I/O waits
- z/OS waits
- IMS internal waits
- Output waits
- External subsystem (ESS) waits

After, you can use bottleneck analysis to statistically estimate the relative impact of each wait reason on the performance for a particular workload.

The results of performance tuning with EPILOG

The goal of tuning is to improve throughput or response time, which requires the issues in a bottleneck profile to be corrected. EPILOG makes it easy to do before-and-after comparisons.

EPILOG provides a workload or resource profile before a change, for example, an average hour, day, or shift. Another profile is provided after the change. You can examine the two profiles on the same screen in a side-by-side comparison.

Chapter 2. EPILOG collector operation

The following topics describe the operation of the EPILOG components, and explains how to operate the EPILOG collector.

EPILOG operational components

EPILOG consists of several components.

Interface
Receives control from z/OS and then starts the collectors.

EPILOG collector
Monitors IMS resource usage, coordinates the operations of the response time analysis (RTA) and bottleneck analysis (DEXAN) collectors, and records the gathered data in the EPILOG data store (EDS).

RTA data collector
Monitors IMS transactions and maintains response-time data.

Bottleneck analysis (DEXAN) data collector
Samples the execution states of the transactions that flow through IMS.

Interface overview

The KOIIA00 interface accepts commands either from a member of its parameter library or from the console of a z/OS operator.

With these commands you can complete the following tasks:

- Start another OMEGAMON for IMS component to monitor the same IMS.
- Display all programs currently running under the interface.
- Stop any program that is running under the interface.
- Obtain a short description of all interface functions.
- Obtain information about virtual storage usage in the interface address space.

If you are not responsible for the operation of the EPILOG collector, you can skip to the Chapter 3, “EPILOG reporter operation,” on page 19 topic.

Interface operation

Normally, you do not need to issue commands to the interface. Sufficient defaults are built into the system so that all necessary actions are taken automatically when you start the interface. Be familiar with the functions and features of the interface.

The interface (KOIIA00) runs in its own address space. Not only the interface but also the three collectors are isolated from the IMS address space. They do not use any virtual storage space within the IMS control region’s private area.

The startup parameter to KOIIA00 determines which IMS system to monitor based on the IMSID. When you start the interface, it checks the dispatching sequence of address spaces to ensure that the interface precedes IMS. If it does not, the interface issues a warning message to the console and continues to work. Collector values can be accurate only if the interface has a higher priority than IMS. (See the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide* for a description of the necessary JCL.)

The interface has two z/OS MODIFY IDs; one is a z/OS ID that detects when IMS ends, and the other is an internal z/OS ID.

1. The first ID can be addressed by the z/OS operator console and can accept z/OS STOP and MODIFY commands. These commands terminate the OMEGAMON for IMS address space.

2. The internal ID accepts only z/OS MODIFY commands from a z/OS operator console, not z/OS STOP commands.

The interface creates an internal ID for itself by prefixing two characters to the IMSID of the IMS system it is monitoring. It notifies z/OS that it accepts z/OS MODIFY commands addressed to that ID. The default prefix is M0. If you run your IMS control region with an IMSID of IMSA, the interface asks z/OS to pass it all z/OS MODIFY commands entered for the ID of M0IMSA. The interface processes only z/OS MODIFY commands; it rejects z/OS STOP commands for the internal ID and displays an error message.

When the interface starts, it automatically runs a series of commands. These commands are in member KOI*mp*P00 in the RKANPAR library. You receive a default series of commands that you can modify. (See the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.)

The interface issues its first command itself to the default startup member, so that its operation can be automatic:

```
EXEC KOImpP00
```

The startup member automatically processes only at initial region startup.

After the interface completes all the actions that are defined in KOI*mp*P00, you can issue more interface commands through the z/OS MODIFY facility by using the OMEGAMON for IMS internal ID.

The standard OMEGAMON for IMS installation procedure requires you to start the EPILOG collector automatically. To start the EPILOG collector, remove the comment character from the EXEC KEICOL*mp* command in the default KOI*mp*P00 provided.

The following is an example of a **MODIFY** command that uses the internal ID and default prefix:

```
MODIFY  M0IMSA,START EPILOG
```

or

```
F M0IMSA,START EPILOG
```

When you start the interface, the display active command **D A,L** (a system command) displays one line that contains identifiers for the IBM monitor job and another line that shows identifiers for the internal MODIFY ID used by the interface. Suppose the IMS you are monitoring runs as a batch job, with an IMSID of IMSA and JOBname MONIMSP. The **D A,L** command displays:

```
JOBname    STEPname   PROC STEPname
MON210T    IMS        IMSPROD        NSW   J   <JOBname>
M0IMSA     M0IMSA     MONIMSP        NSW   J   <MODIFY ID>
```

If you start the interface as a *started task*, modify the STEPname to produce a unique z/OS MODIFY ID. For example, IMSMON:

```
START OMIMS.IMSMON
```

Now **D A,L** shows:

```
JOBname    STEPname   PROC STEPname
OMIMS      IMSMON     IMSPROD        NSW   S   <JOBname>
OMIMS      M0IMSA     IMSMON         NSW   S   <MODIFY ID>
```

Remember, in both these examples the second line is the internal MODIFY ID that the interface builds.

In this example, you can terminate the interface with one of the following MODIFY or STOP commands to the z/OS ID:

```
STOP IMSMON
P IMSMON
MODIFY IMSMON,STOP
F IMSMON,STOP
```

The interface operates automatically and usually does not require operator intervention.

Interface commands

You can issue commands directly in the console or store the commands in an RKANPAR member. From the console, you must use the **MODIFY** command before you can issue any other commands from the console. To store commands in an RKANPAR member, you must issue the **EXEC** interface command.

See Table 1 on page 9 for a description of the interface commands for the EPILOG historical component.

Table 1. Interface commands

Command	Description
*	Indicates a comment and must begin in column 1. The interface ignores characters after the *.
DISPLAY	Displays active interface subtasks
EXEC	Runs the commands in the specified member
HELP	Displays help for all or specific interface commands
IF	Conditionally processes EXEC, START, or STOP commands
LIST	Displays active interface subtasks (alias of DISPLAY)
LOG	Sends a message to the z/OS console
P	Stops an interface subtask (alias of STOP)
S	Starts an interface subtask (alias of START)
START	Starts an interface subtask
STOP	Stops an interface subtask
VARY	Changes the option member that is used by the EPILOG collector

The Comment (*) command

The comment command places comments in the members of the RKANPAR library.

You must place the asterisk (*) in the first column. Indicate a continuation of the comment by placing a nonblank character in column 72. An example of a comment line in an EXEC member follows:

```
* THIS COMMENT CAN SAY ANYTHING YOU WANT.
```

DISPLAY or LIST command

The DISPLAY or LIST command displays all tasks that are currently active. An internal ID displays along with the program name of the task. You can use the ID with the STOP command to change the status of the task.

The format of the DISPLAY or LIST command is as follows:

```
DISPLAY
LIST
```

The output from the DISPLAY or LIST command is as follows:

```
OIR043: EPILOG - THE FOLLOWING TASK IDS ARE ACTIVE:
OIR044    ID=RT        PROGRAM=RI310310
OIR044    ID=DC        PROGRAM=DC310310
OIR044    ID=DX        PROGRAM=DX310310
OIR044    ID=EP        PROGRAM=EI310310
```

The EXEC command

The EXEC command processes a predefined set of commands that are stored as a member in the RKANPAR library. Issue a MODIFY command or as a command in an EXEC member. The nesting limit of EXEC commands that are processing by a single command invocation is 10. This nesting limit prevents EXEC loops where A runs B and B runs A.

The format of the EXEC command is as follows:

```
EXEC <member_name>
```

You can think of an EXEC set of members as a JCL procedure or as a TSO CLIST. The EXEC member can contain commands such as START and STOP, or even another EXEC command. When one EXEC command processes inside another EXEC member, it is as if all the commands in the second EXEC member replace the calling command in the first EXEC member, the calling command.

For example, if you enter the following command at the console:

```
F M0IMSA,EXEC MEMBERA
```

And if MEMBERA contains the following commands:

```
START EPILOG
EXEC  MEMBERB
START OMEGAMON,UNIT=53E,...
```

And MEMBERB contains the following commands:

```
LOG *** OM/IMS VTAM interface START - APPLID=OIVTAM ***
START OIVTAM,APPLID=OIVTAM,UMAX=05
```

The effect of EXEC MEMBERA is the same as if you entered the following commands for M0 at the console:

```
F M0IMSA,START EPILOG
F M0IMSA,LOG *** OM/IMS VTAM interface START - APPLID=OIVTAM ***
F M0IMSA,START OIVTAM,APPLID=OIVTAM,UMAX=05
F M0IMSA,START OMEGAMON,UNIT=53E,...
```

The HELP command

The z/OS console operator can use the HELP command with no operand to determine what commands the interface supports.

The format of the HELP command with no operands is as follows:

```
F M0IMSA,HELP
```

The output from the HELP command is shown in Figure 2 on page 11:

```
LOG ‘HELP’ Command
 Syntax: HELP <command-name>

 Description:  The ‘HELP’ command displays the help information available
 on the commands that control the EPILOG interface.

 HELP is available for all the commands below:

 *         - Comment (ignored by the interface)
 EXEC      - Executes the commands in the specified member
 DISPLAY   - Displays active interface subtasks
 HELP      - Displays help for all or specific interface commands
 LIST      - Displays active interface subtasks (alias of DISPLAY)
 LOG       - Sends a message to the z/OS console
 RESTART   - Terminates an existing subtask and restarts it
 START     - Starts an interface subtask (e.g., EPILOG)
 STOP      - Stops an interface subtask
 VARY      - Changes the option member used by the EPILOG collector
```

Figure 2. HELP command output

If you are a master console operator, you usually use the HELP command to display information about any interface command. If you do not specify a command name, if the command is unrecognized, or if you type in HELP, the information for HELP displays on the console. The format to obtain HELP for a specific command (in this case, START) is:

```
F M0IMSA,HELP START
```

The IF command

The IF command conditionally processes the EXEC, START, or STOP command that follows it in the same command input. The IF command is especially useful if you want to use the same members of the RKANPAR library to control copies of EPILOG which run with different IMS systems.

The IF command is a way to test several different values to determine whether the command that follows its THEN keyword runs. The format of the IF command is shown as follows:

```
IF <IMSID=cccc>          Then <EXEC  ...>
   <SMFID=cccc>               <P     ...>
   <CPUID=cccccccccccc>       <S     ...>
                              <START ...>
                              <STOP  ...>
   <IMSTYPE=ccc>              <.........>
```

Figure 3. IF command format

Table 2 on page 11 describes the IF test parameters and their associated values.

Table 2. IF test values

Parameter	Value	Description
CPUID	(12)c OR (6)c	The 12-character hardware CPUID of the machine. See title line on first page of a memory dump. -OR- The 6-character hardware processor serial number of the machine. It is available from the RMF CPU report. For a multiprocessor, the CPUID compares with all those within the multiprocessing complex.
IMSID	cccc	IMS ID of the IMS system you are measuring.
IMSTYPE	CTL DBC	Determines the set of interface commands that run, depending upon whether the IMS environment is a DB/DC (CTL) environment or a DBCTL (DBC) environment.

Table 2. IF test values (continued)		
Parameter	**Value**	**Description**
SMFID	cccc	SMF ID of the z/OS system you run. Can be found in SYS1.PARMLIB(SMFPRMnn).

The output that results from the IF command depends on whether the tests are successful or not. If a test is successful, you receive the normal messages for the command that is processed. If the test fails because of an invalid value in the command, you receive a message indicating this.

The LOG command

The LOG command displays a message at the system console. This command is for use in the EXEC members to indicate the commands that you set to process in that member.

The format of the LOG command is as follows:

```
LOG *** This message will be sent to the system console ***
```

The output from the LOG command produces output that is the same as its input. A typical use is to indicate the name of the member that is processed, for example:

```
LOG *** Processing EMVTAM:hp1.mp:ehp1. ***
```

Another typical use is to indicate the start of a task such as the EPILOG collector, for example:

```
LOG *** Starting EPILOG COLLECTOR SUBTASK ***
```

The P or STOP command

The STOP command stops any interface subtask. To end EPILOG collector operation, issue a STOP command to the interface.

The system console operator enters the STOP command by using the z/OS MODIFY command. The STOP command needs a task ID to know which task to stop. This ID can be found by using the DISPLAY or LIST command.

The format of the STOP command is as follows:

```
F M0IMSA,STOP ID=EP
```

The output of STOP consists of task termination messages followed by a task detached message. If the task does not promptly process the interface termination request, the interface is forcibly detached.

The S or START command

The START command starts the EPILOG collector, the RTA collector, and the DEXAN collector tasks.

A sample START command for EPILOG for your installation is in *rhilev.midlev*.RKANPAR(KEICOL*mp*). Here is the format for the START command:

```
START EPILOG,OPTION=cccccccc
```

where *cccccccc* is the name of the member that contains startup options and collector group definitions.

Note: Issuing the START EPILOG command automatically starts the RTA and DEXAN collectors.

Important: If you run OMEGAMON for IMS's EPILOG historical component during the same IPL of your system as the predecessor product, EPILOG for IMS, Version 320 or earlier, you must specify the CMPAT=YES parameter on the START EPILOG command.

```
START EPILOG,CMPAT=YES
```

The VARY command

During collection (after EPILOG starts), you can change the options.

Use the following command to change the options:

```
F     M0IMSA,VARY ID=EP OPTION=member name
```

If you issue this command, the new options take effect at the end of the current interval, that is, the interval keyword as specified in the previous option member.

Options for the EPILOG collector

You communicate with the EPILOG collector through a data set member that is defined by the RKANPAR DD card. Options in the data set member define the various parameters that control EPILOG collection. Each option has a default setting. You can change an option in freeform style by using the syntax keyword.

You can specify options at startup or change them during a collection interval. If you choose to change options, your new settings become active at the beginning of the next collection interval (not during the current interval). The RKANPAR data set in which you store your options is defined with a record length of 80. The collector reads significant data only from the first 72 bytes of each record. Data that is entered after column 72 is ignored. Do not specify a record length other than 80 or you might get highly unpredictable results.

EPILOG collector keywords

Use the following keywords to define options for the EPILOG collector.

The following keywords are described in this topic.

- INTERVAL
- DEGCYCLE
- DBIO
- SMFDATA
- NOSMFDATA
- SMFNUM (or SMFN)
- SYMNAME
- ALTDATA
- MESSAGES (or MSGS)
- NOMESSAGES (or NOMG)
- NORESC

The terms in this list are described in "Data collection groups" on page 17.

- GROUP
- GROUP Operands:
  - TRAN
  - PSB
  - CLASS (or CLS)
  - TERM
  - NODE

INTERVAL
: This option sets the interval at which EPILOG collects, combines, and writes data to the EDS and SMF. The range of this interval can be from 10–120 minutes (2 hours). Valid options are RMF or a number of minutes. The default is the RMF cycle or 15 minutes if RMF is not installed or is not active.

DEGCYCLE
This option sets the cycle at which EPILOG collects bottleneck data. The range of this cycle is expressed in 100ths of a second and extends from 50/100–1000 (.5–10 seconds). The default is 500/100, or 5 seconds.

DBIO
Bottleneck analysis is normally performed for all databases. If YES is specified for this option, EPILOG collects statistics for each individual active database. On a system with many active databases, the storage requirements for EPILOG database collection may be prohibitive. If this is the case, EPILOG sends a WTO message to the system console, and it stops individual database collection. Valid entries for this keyword are YES and NO. The default is NO.

SMFDATA
EPILOG writes the collected data to the EDS at the end of every collection interval. By default, it also writes to SMF. If you want to write to SMF use this keyword.

NOSMFDATA
This option causes EPILOG *not* to write to SMF.

SMFNUM
When EPILOG writes to SMF, it must have a way to separate this data from the data that SMF normally collects. The SMFNUM option allows you to set a number from 0–255 that represents the EPILOG Id for collected data. Make sure you choose a record number that is not already in use. The default is 185. (Short form is SMFN.)

CAUTION: Other IBM products (IBM Tivoli® OMEGAMON Classic, OMEGAMON Classic for z/OS, EPILOG for IMS, and DELTAMON® for z/OS) also write records to SMF. If any of these products is installed, be sure that IBM Tivoli OMEGAMON for IMS does not use an SMF record number that is already assigned. Default numbers for these IBM products are:

180
OMEGAMON for IMS and its predecessor EPILOG for z/OS

182 and 183
DELTAMON for z/OS

185
OMEGAMON for IMS and its predecessor EPILOG for IMS

SYMNAME
The SYMNAME command lets you define a table of 8-character symbolic names for the numeric groups defined with the GROUP command. SYMNAME accepts a set of operands in parentheses, containing a symbolic name (optionally enclosed with single quotes to allow imbedded blanks) followed by an associated group number. Each symbolic name must be unique, and each group can be defined by only the symbolic name. A symbolic name cannot be all numeric.

```
SYMNAME (PAYROLL 1)
```

```
SYMNAME (BILLING 2)
```

```
SYMNAME (‘TEST 1’ 14)
```

You can let EPILOG set default symbolic group names for all groups you define, which are defined as GROUP *nn*, where *nn* is the number of the group.

When using the DISPLAY command in the EPILOG reporter to display information about a group, you can refer to the group by either its number (with GRP) or the symbolic name that you assigned (with SYMNAME).

ALTDATA
Used to collect resource data when another vendor product is used instead of RMF to monitor system resources.

MESSAGES (or MSGS)
This option allows EPILOG to display to the z/OS operators console of the following messages: ECW002, ECW004, and ECW006. This is the default.

NOMESSAGES (or NOMG)
This option suppresses the writing to the z/OS console of the following messages: ECW002, ECW004, and ECW006.

NORESC
This option disables resource collectors. See *thilev*.RKANPAR(KEIOPT00) for an example of NORESC.

The syntax for NORESC (NO RESource Collector) is as follows:

```
NORESC(ccc)
```

or

```
NORESC(ccc,ccc)
```

where *ccc* is the name of the resource collector (the last 3 characters of the Resource Display Rccc keyword). For example, the parameter NORESC(TXU,DAS,VSP) disables collectors RTXU, RDAS, and RVSP.

There is one exception to this: the RDDN resource collector is RDN, rather than DDN. Use NORESC(RDN). See "Resource displays (Rccc keywords)" on page 32 for more information.

The following figure shows how these defaults are specified.

```
************************************************************************
*                                                                      *
*              EPILOG COLLECTOR SAMPLE OPTION MEMBER                   *
*                                                                      *
************************************************************************
*
*                      ** RULES **
*
*     STYLE IS FREEFORM.
*     DELIMITERS ARE ( , ) AND SPACE
*     ACCEPTABLE CHARACTERS ARE IBM ALPHAMERIC
*     * - ASTERISK INDICATES COMMENT IN FIRST COLUMN,
*         GENERIC CHARACTER IN OTHER THAN FIRST COLUMN.
*     SPACES ARE ALLOWED ANYWHERE
*     RESTRICTION - KEYWORD ITSELF CANNOT SPAN LINES
*
*
*              ** SAMPLE OPTION MEMBER FOLLOWS **
*
*
********************************************************************
**     INTERVAL IS PERIOD IN WHICH EPILOG COLLECTS AND WRITES.   **
**     INTERVAL CAN BE EITHER RMF OR NUMBER OF MINUTES (10-120). **
********************************************************************
       INTERVAL  (   RMF)

********************************************************************
**     DEGCYCLE IS BOTTLENECK ANALYZER STIMER INTERVAL.           **
**     DEGCYCLE IS IN HUNDREDTHS OF SECONDS AND CAN BE 50-1000.  **
********************************************************************
                                            DEGCYCLE(500)
********************************************************************
**     BLANK LINES ARE ALLOWED ANYWHERE.                          **
********************************************************************
********************************************************************
**     DBIO IS YES OR NO - COLLECT BOTTLENECK STATS BY DATABASE  **
********************************************************************
 DBIO(NO)

********************************************************************
**     SMFDATA IS YES OR NO - WRITE EPILOG RECORDS TO SMF.        **
********************************************************************
NOSMFDATA

********************************************************************
**     SMFNUM IS A VALID USER SMF RECORD NUMBER.                  **
********************************************************************
              SMFNUM(185)
```

Figure 4. Sample collector option member

```
******************************************************************
**                                                              **
**     GROUPS ARE FOR COLLECTION PURPOSES AND ARE # 1-30         **
**       TRAN IDENTIFIES A TRANSACTION IN THE GROUP             **
**       PSB IDENTIFIES A PSB IN THE GROUP                      **
**       CLASS IDENTIFIES A SHEDULING CLASS IN THE GROUP        **
**       TERM IDENTIFIES LTERMS IN A GROUP - *NOTE* GROUPS      **
**       DEFINING LTERMS CANNOT DEFINE TRANS, PSBS AND CLASSES.  **
**       "*" ARE USED TO IDENTIFY GENERIC NAMES                 **
**                                                              **
******************************************************************

  GROUP(1)  TRAN(TRAN1 ,TRAN2 TRAN3) PSB(PSB1)      TRAN(TRAN4)
            TRAN(     TRAN5         TRAN6
               TRAN7 )
  GROUP(2)  TERM(LTERM1) TERM(LTERM2)
            TERM(LTERM3
                   LTERM4)
  GROUP(3)
            CLASS(1 4 5    ) PSB( PSB2,PSB3)
            TRAN( TR*)   PSB(PSB* )
  GROUP(4)
            TERM(TERM* )

  GROUP(5)  NODE(LU1234,LU123*,LU*)

    SYMNAME(ACCOUNTS 1)
    SYMNAME(PAYROLL 3)
    SYMNAME(ETEGRP 5)
```

Figure 5. Groups screen

Note: For information about the remaining terms and using wildcards, see the following topics.

- "Data collection groups" on page 17
- "Wildcards" on page 18

Data collection groups

Transaction or terminal groups target the information that the bottleneck and response-time collectors collect. You can target service-level analysis to a set of similar applications or terminals, so that your tuning efforts can focus on the specific bottlenecks that impede your system. With EPILOG, you can define 30 different groups.

The GROUP keyword defines transaction, terminal, and VTAM® LU groups. If you do not define any groups, EPILOG uses a default of 10 groups (1–10) with 10 corresponding scheduling classes (1–10).

There are three types of groups:

- LTERMs only
- Any combination of transactions, PSBs, and scheduling classes
- VTAM nodes

The three group types are mutually exclusive; you cannot mix them.

Use any of five operands under the GROUP keyword to specify the composition of the groups. These five operands and their syntax are as follows:

TRAN
: Transaction code

PSB
: PSB names

CLASS
: Scheduling class number (1-255)

TERM
: Terminal names

NODE
LU sessions

The parameters for any operand data must be in parentheses and each defined name must be separated by delimiters. Valid delimiters are spaces and commas. You can specify up to 255 entries between the set of parentheses.

```
TRAN (GEORGE,TRAN1 PAYROLL EMPLOY,SALES)
```

The preceding operand (TRAN) names five different transactions. Notice that the commas and spaces mean the same within a set of parentheses; you can choose to use either or both.

Operand names can use from 1–8 alphanumeric characters (including three special characters: #, $, and @). A name can start with an alphabetic character, a number, or one of the special characters. Only transactions, PSBs, and LTERMs that are defined to IMS are defined in groups.

Wildcards

If you use an asterisk (*) as a character in an ID name, it functions as a wildcard. Any character can fill the asterisk (*) position, which is useful when you define transactions to specific groups.

For example, you can enter the following command:

```
TRAN (GEORGE,TRAN* PAY* E***OY,*ALES)
```

The group now includes transaction GEORGE, any transaction that begins with TRAN, any transaction that begins with PAY, any transaction that begins with E and has O and Y in the fifth and sixth positions, and any transaction that has A, L, E, S in the second through fifth positions.

Trailing asterisks (TRAN* and PAY*) match all remaining characters. For example, entering TRAN*** for TRANSAC is the same as entering TRAN*. The same naming convention holds for terminal names and transaction names:

```
TERM (L*)
```

The terminal names include all those terminals that begin with L.

Setting options and defining groups

You can communicate options and define groups to the collector at two times: at startup time, or during collection.

If you want to set options at startup time, specify the OPTION keyword with the START command. If no OPTION= is specified, EPILOG uses the defaults. For example:

```
START EPILOG OPTION=member name
```

The member name is the name of the data set member in which the options are kept. The RKANPAR DD card in the OMEGAMON for IMS procedure defines the PDS, which contains the specified member.

During collection, you can change the options with the VARY command. See "The VARY command" on page 13.

Chapter 3. EPILOG reporter operation

The commands to issue to the reporter and the kinds of information the reporter displays are described in the following topics.

EPILOG reporter modes of operation

The reporter is a central feature of EPILOG; it displays information from the EPILOG data store (EDS). You can start the reporter in batch, under TSO (in two modes), or under ISPF (in three modes).

Table 3 on page 19 describes these modes of operation.

Table 3. EPILOG reporter modes of operation

Mode	How it operates	Comments
Batch	Reads a set of commands in a data set pointed to by the RKEI2IN DD statement. Output displays in a data set which the RKEIOUTR DD statement points to. Messages are in the RKEIOUTM data set.	EPILOG includes a sample batch procedure along with input for some sample reports. The *Configuration and Customization Guide*, describes these.
TSO Fullscreen	A CLIST allocates the necessary files and CALLs the EPILOG load module.	In this mode, EPILOG performs fullscreen I/O and provides for scrolling of displayed data. The *thilev*.TKANSAM(KEICLIST) data set invokes the reporter in this manner.
TSO Batch Function	Requires either the RIEI2IN or RKEIOUTR filenames. EPILOG operates as any other batch program which executes under TSO and thus does not do fullscreen TGETs and TPUTs.	This is the normal TSO conversational mode; when the screen is full, three asterisks display to prompt you to press Enter in order to see the rest of the output.
ISPF split-screen	CLISTs use ISPF panels to define the necessary PF keys, including those used to perform ISPF SPLIT and SWAP commands, and invoke the EPILOG load module by CALLing an interface module named EBSPFDR.	This mode allows you to use a second ISPF session while executing the EPILOG reporter.

The *IBM OMEGAMON for IMS on z/OS: Historical Component (EPILOG) User's Guide* contains a complete explanation of EPILOG operations under TSO and ISPF split-screen modes.

Starting the EPILOG reporter

The command that you enter depends on whether you are running EPILOG in ISPF mode, TSO mode, or batch mode.For more information about starting the reporter, refer to *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

Follow these steps to start the EPILOG reporter:

1. Enter the command to start the EPILOG reporter.

   After you issue the command that starts the reporter, the first screen you see is the copyright screen, which looks similar to Figure 6 on page 20.

```
 Copyright 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1991, 1992, 2001
                   IBM corporation. All rights reserved.
                     Use permissible by license only.

                       EPILOG/IMS         Version 530
+==============================================================================+
|                                                                              |
|                        Summary Data Store Statistics                         |
|                                                                              |
|        Date and Time Range               Util    Xt                         |
|        ------------------------------    ----   ---                         |
|   1  TDIMS.I5530SMP.RTENAME.IC1C.RKEIEDS                                     |
|        05/29/15 09:00 - 08/31/15 16:00    98%    18                         |
|                                                                              |
|                                                                              |
|                           Press ENTER to Continue                            |
|                                                                              |
+==============================================================================+
```

Figure 6. Copyright screen

This screen contains copyright information that safeguards IBM's proprietary rights to OMEGAMON for IMS and this product component. Information follows that identifies the EPILOG data store that is defined for the current reporter session, along with statistics for the data store such as the creation date and time, the date and time the EDS was last updated, the percentage of usage, and the number of extents.

2. Press Enter.

   The next panel shows the EPILOG default program function (PF) key settings as shown in Figure 7 on page 20.

```
EPILOG/IMS V530  08/31/15 16:12  Mode: PAGE            1 of 2   PFK DISPLAY
CMD==>
*******************************************************************************

  PROGRAM  FUNCTION  KEYS

PFK  1 'HELP'
PFK  2 'HELP'
PFK  3 'BACK'
PFK  4 'CONTROL RECALL'
PFK  5 'CONTROL TITLE'
PFK  6 'DIS RALL TODAY COMBINE STIME(9) ETIME(17)'
PFK  7 'CONTROL SCROLL UP'
PFK  8 'CONTROL SCROLL DOWN'
PFK  9 'CONTROL LOG'
PFK 10 'DIS SYSTEM TODAY SUMMARY COMBINE(1H)'
PFK 11 'DIS GRP(1) TODAY RIF(R0T(>1S)) AUTO'
PFK 12 'PFK'
PFK 13 'HELP'
PFK 14 'CONTROL MODE'
PFK 15 'BACK'
PFK 16 'CONTROL RECALL'
PFK 17 'CONTROL TITLE'
PFK 18 'DIS RALL TODAY COMBINE STIME(9) ETIME(17)'
PFK 19 'CONTROL SCROLL UP'
PFK 20 'CONTROL SCROLL DOWN'
PFK 21 'CONTROL LOG'
PFK 22 'DIS SYSTEM TODAY SUMMARY COMBINE(1H)'
PFK 23 'DIS GRP(1) TODAY RIF(R0T(>1S)) AUTO'
PFK 24 'PFK'
```

Figure 7. Default program function key settings

This screen lists the default program function key assignments. For instructions on changing these assignments, refer to the *IBM OMEGAMON for IMS on z/OS: Historical Component (EPILOG) User's Guide*.

EPILOG screen

You communicate with EPILOG through commands and their associated keywords and options.

"EPILOG reporter commands and syntax guidelines" on page 22 describes these commands. When you run under TSO, you enter EPILOG commands on the **CMD==>** line. For example, here is the screen that is displayed when you start your EPILOG session:

```
EPILOG/IMS V530  08/31/15 16:12  Mode: PAGE                     ENTER COMMAND
CMD==>
******************************************************************************
```

Figure 8. Starting your EPILOG session

EPILOG input errors

EPILOG produces appropriate error messages when a command string is invalid or ambiguous in some way.

For example, the command that is shown in the following example contains a group number, which is longer than the maximum of three digits:

```
EPILOG/IMS V530  08/31/15 16:12  Mode: PAGE                     ENTER COMMAND
CMD==> DIS GRP(9999)
******************************************************************************
```

Figure 9. Exceeding the maximum length of a group number

When you press Enter, EPILOG positions the cursor at the location of the first error (if known) and displays the number of the first error message in the warning area at the upper right of the screen. A ruled measurement line displays after the command line, followed by all the error messages.

```
EPILOG/IMS V530  08/31/15 16:12  Mode: PAGE                     ERROR EI857
CMD==> DIS GRP(9999)
       ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
**** EI857: Invalid Transaction Group Number
**** EB490: No Workload, Resource, or display type was supplied
******************************************************************************
```

Figure 10. Error messages

To correct the error, alter the command line as necessary and press Enter again.

Issuing TSO commands

You can start TSO CLISTs and command processors directly from EPILOG by issuing the TSO command. For example, to issue a list catalog command while EPILOG is running, enter `TSO LISTC`.

Because the EPILOG reporter does not require APF authorization, it is not possible to run APF-authorized programs underneath it.

This feature is subject to the following restrictions and cautions:

- You cannot use the `TSO TEST` command.
- The TSO command is not valid when EPILOG is running as a batch program.
- Do not run any TSO commands that alter characteristics of resources that support EPILOG (for instance, help text, parameters, data stores, or storage).

EPILOG reporter commands and syntax guidelines

The EPILOG collector gathers information about response times, bottleneck, and resource usage and writes this data to SMF and an online EPILOG data store. To display this information about a TSO terminal or send it to a SYSOUT file in batch, use the reporter DISPLAY command.

The reporter displays information in the database in various formats, depending on the commands and keywords you enter. It accepts commands and keywords in any order.

EPILOG reporter command syntax

EPILOG uses essentially the same syntax conventions as TSO. You separate a keyword from either an argument or another keyword by using one or more blanks, equal signs, commas, or parentheses. You must pair parentheses, and you cannot begin an input line with a closing parenthesis. For consistency, the following topics show only parentheses as separator. We suggest that you use a similar convention.

Enclose arguments that contain blanks or other separators in single quotation marks ('). Do not continue an argument that is enclosed in quotation marks from one line to the next.

Most of the keywords have both a long and a short form. Either one is valid. You can mix long and short forms within a command. EPILOG recognizes most keywords if you enter an unambiguous abbreviation of the long form (such as COMBI instead of COMBINE).

If you must enter a command that is too long to fit on one line, end the first line with a hyphen surrounded by blanks (-). The hyphen is called the continuation character because you can continue the command on the next line. A command line with the continuation character at the end does not run. Instead, you see all the command segments that are entered with a prompt for more input. You can use as many lines as necessary to enter the command. You can enter comments after the hyphen, because any characters on a line after the hyphen are ignored.

For example, the following diagram shows the screen that displays when you enter the string `DIS GRP(1) -` , where DIS is an abbreviation for DISPLAY. Because you specified the hyphen continuation character, EPILOG prompts you for the rest of the command.

```
EPILOG/IMS V530  08/31/15 11:29  Mode: PAGE                     CONTINUE INPUT
CMD==>
******************************************************************************
DIS GRP(1) -
```

Figure 11. DIS screen with continuation

To enter a comment line, place an asterisk (*) in the first column of the command line. The comment writes to the message data set (allocated to RKEIMSG).

The following table introduces the reporter commands and indicates which topic describes each command in detail.

Table 4. EPILOG reporter commands

Command	Short	Description	Topic
DISPLAY	DIS	Displays information in the EDS (or SMF) on a TSO terminal or sends it to a SYSOUT file in batch mode.	"DISPLAY command categories" on page 23
SET		Sets the default options for the DISPLAY command.	"Keyword default setting (SET command)" on page 39
CONTROL	CNTL	Specifies the colors, paging, and other options for 3279-type terminals.	"Full-screen control (CONTROL command)" on page 41

Table 4. EPILOG reporter commands (continued)

Command	Short	Description	Topic
PAGESEP		Specifies options for special display requirements when you use EPILOG as a batch reporter.	"Report separators (PAGESEP command)" on page 47
DATASTOR		Allocates and deallocates data stores during a reporter session.	"Dynamically allocating and deallocating datastores (DATASTOR command)" on page 44
INQUIRE	INQ	Displays information about the data store's status and contents.	"Displaying EDS status (INQUIRE command)" on page 45
REOPEN		Changes the number of VSAM and index buffers.	"Specifying index and data buffers (REOPEN command)" on page 46
CREATEM	CMAT	Creates a new automatic analysis member.	"Create an A-matrix (CREATEM keyword)" on page 75
DELETEM	DMAT	Deletes an automatic analysis member.	"Delete an A-matrix (DELETEM keyword)" on page 76
LISTM	LMAT	Lists an automatic analysis member.	"List an A-matrix (LISTM keyword)" on page 76
REPLACEM	RMAT	Modifies and replaces an automatic analysis member.	"Replace an A-matrix (REPLACEM keyword)" on page 76

DISPLAY command categories

The DISPLAY command (also known as DIS) has five categories of keywords.

The syntax of the DISPLAY command is as follows:

```
DISPLAY  display type  -
         workload  -
         date/time  -
         exceptions  -
         miscellaneous options
```

Figure 12. DISPLAY command syntax

display type
: EPILOG produces five different types of data displays: DETAIL, GSUM, SUMMARY, resource, and COLLECTOPT. For more information, see "Display types" on page 24. DETAIL is the default.

workload
: The DETAIL and SUMMARY displays require that a workload be specified. A workload is a transaction group that allows you to focus your tuning efforts to specific transactions. For more information about workload specifications, see "Workloads" on page 34.

date/time
: The date and time specification limits the display to a particular set of time intervals. If you did not supply date/time information, EPILOG tries to process the entire set of online data, beginning with the oldest and ending with the most current. The time specifications are flexible; see "Time specifications" on page 36.

exceptions

Part of the IBM tuning technique involves selecting only those intervals during which the service goals were not achieved. Exceptions are used with DETAIL, SUMMARY, and GSUM displays so that you can select only specific kinds of intervals, such as `all intervals with scheduling waits greater than 2 seconds` or `all transactions which waited for database access for more than 1 second`.

For more information about how to use this keyword, see Chapter 5, "Selecting EPILOG displays based on bottleneck factors," on page 55.

miscellaneous options

Each display type has its own array of optional keywords and parameters. See Chapter 8, "Miscellaneous keywords in EPILOG reports," on page 81.

Display types

EPILOG contains eight different display types. The following table provides information about the displays.

Table 5. Display types

Keyword	Short	Description
DETAIL	DTL	Display detailed degradation and response time for a workload (individual collection intervals).
SUMMARY	SUM	Display main execution state and response time for a workload by interval.
GROUPSUMMARY	GSUM	Displays transaction rate and average response time by transaction group.
Resource (R*ccc*)		Resource panel keywords: RALL, RBUF, RCIP, RCFP, RCMP, RCPU, RDAS, RDBE, RDDN, RDEV, RDMP, RECA, RFPI, RFPO, RFPP, RFPR, RFRP, RINF, RLAT, RLOG, RMFP, ROSP, RPAG, RPIP, RPSP, RQBP, RSAP, RSCD, RSRM, RTXU, RVSP (See Table 6 on page 32) There is one exception to using *ccc*. The RDDN resource collector is RDN, rather than DDN.
COLLECTOPT	COPT	Display collector options and GROUP definitions.

All display types are mutually exclusive.

Detailed workload (DETAIL keyword)

DETAIL is the default display type, which decomposes the significant execution states and performance bottlenecks of an average transaction within the workload for a single collection interval. Intervals are usually the same as the system RMF interval. You can synchronize EPILOG intervals with RMF intervals when you install the product.

When you use DETAIL, you cannot use the SUMMARY keyword. Figure 13 on page 25 shows an example of the DETAIL report. (This display looks different in a DBCTL environment; see " DETAIL display in a DBCTL environment" on page 29 for more information.)

```
+-----------------------------------------------------------------------------+
| Workloads:    Transaction group 3                                           |
| Periods:      Starting on 08/29/15 at 10:45 and Ending on 08/29/15 at 11:00 |
| Misc Options: Detail PLOTMIN(3)                                             |
|                                                                             |
|_____________________________________________________________________________|
| TRANSACTION GROUP =   3                               Symbolic Name=cccccccc |
+=============================================================================+
| PERIOD: 10:45 to 11:00 on 08/29/15                Elap =   15:00 M  IMSD   |
+=============================================================================+
|                            RESPONSE TIME DATA                               |
+=============================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue        |       0.74s       |        79       |        5.27        |
|Pgm Input Queue    |       0.33s       |        78       |        5.20        |
|Processing         |       8.78s       |        78       |        5.20        |
|Response Time 0    |       9.85s       |        78       |        5.20        |
|Output Queue       |       0.10s       |        76       |        5.07        |
|Response Time 1    |       9.99s       |        76       |        5.07        |
+=============================================================================+
|                               DEGRADATION DATA                              |
+=============================================================================+
|COMPETING_STATES________TIME_____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU              1.77 S   18.0|------->.   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.78) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS   (0.99) S (10.1)|---->   .   .   .   .   .   .   .   .   .|
|Scheduling Waits       .54 S   10.9|---->   .   .   .   .   .   .   .   .   .|
| Wait for MPP        (.54) S (10.9)|---->   .   .   .   .   .   .   .   .   .|
|Database I/O Waits    6.12 S   62.1|------------=============>  .   .   .   .|
| PMFG021            (5.82) S (59.0)|------------==========>.   .   .   .   .|
| BE3INDX            (0.30) S  (3.1)|->  .   .   .   .   .   .   .   .   .   .|
|z/OS Waits             0.44 S   4.5|->  .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP) (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|IMS Activity          0.44 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL   (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|-----------------------------------|                                     .   .|
|AVERAGE TOTAL (Resp Time 0) 9.85 S |                                     .   .|
+=============================================================================+
```

Figure 13. DETAIL report

```
+------------------------------------------------------------------------------+
| Workloads:    Transaction group 3                                            |
| Periods:      Starting on 08/29/15 at 10:45 and Ending on 08/29/15 at 11:00 |
| Misc Options: Detail PLOTMIN(3)                                              |
|                                                                              |
|______________________________________________________________________________|
| TRANSACTION GROUP =   3                              Symbolic Name=cccccccc |
+==============================================================================+
| PERIOD: 10:45 to 11:00 on 08/29/15                 Elap =   15:00 M  IMSD   |
+==============================================================================+
|                            RESPONSE TIME DATA                                |
+==============================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue          |       0.74s       |       79       |       5.27         |
|Pgm Input Queue      |       0.33s       |       78       |       5.20         |
|Processing           |       8.78s       |       78       |       5.20         |
|Response Time 0      |       9.85s       |       78       |       5.20         |
|Output Queue         |       0.10s       |       76       |       5.07         |
|Response Time 1      |       9.99s       |       76       |       5.07         |
+==============================================================================+
|                               DEGRADATION DATA                               |
+==============================================================================+
|COMPETING_STATES________TIME____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU              1.77 S   18.0|------->.   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.78) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS   (0.99) S (10.1)|---->   .   .   .   .   .   .   .   .   .|
|Scheduling Waits       .54 S   10.9|---->   .   .   .   .   .   .   .   .   .|
| Wait for MPP        (.54) S (10.9)|---->   .   .   .   .   .   .   .   .   .|
|Database I/O Waits    6.12 S   62.1|------------============>   .   .   .   .|
| PMFG021            (5.82) S (59.0)|------------===========>.   .   .   .   .|
| BE3INDX            (0.30) S  (3.1)|->  .   .   .   .   .   .   .   .   .   .|
|z/OS Waits             0.44 S   4.5|->  .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP) (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|IMS Activity          0.44 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL   (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|------------------------------------|                                    .   .|
|AVERAGE TOTAL (Resp Time 0) 9.85 S |                                    .   .|
+==============================================================================+
```

Figure 14. DETAIL report continued

The concept of bottleneck analysis, on which this display is built, is explained in the "Bottleneck analysis" on page 5 topic. The display is produced by entering this command with the following keywords:

```
DISPLAY DETAIL GROUP(3) SDATE(08/29/15) STIME(1045) EDATE(08/29/15)
ETIME(1100) PMIN(3)
```

If you used abbreviations, the first three keywords can be DIS DTL GRP(3).

The report describes the workload (transaction group 3), the interval (10:45 to 11:00 on 08/29/15), and the elapsed time (15 minutes).

The rest of the display is described in the following subsections.

Response time data

The middle section of the DETAIL panel shows response-time data on a workload basis. The RTA data collector collects the data. It measures queuing and service times within IMS. RTA data collector response time is the time that it takes to acknowledge an input message from the processing network and to initiate a response. The RTA data collector does not measure the time transactions spend in communications facilities.

A typical IMS transaction proceeds through the following stages:

1. The input message text is received from nodes (terminals, for example) within the communications network.
2. The message is stored in the message queue for subsequent processing until all necessary resources are available to process the transaction.

3. Other messages can be added to the input message queue during the processing of transactions by application programs.
4. When resources are available, an application program receives the input message text from the message queue and issues a DL/I message get unique (GU) call from the I/O PCB.
5. During application processing, output messages are prepared for various system destinations. Some messages are directed to other applications within the system. These messages go to the input message queue as described in item 3. The remaining messages go to communications destinations (for example, terminals) within the network. These output messages are also stored in the message queues.
6. After the application program processes the transaction, output messages are taken from the message queues for delivery to final destinations. Output queue times can be lengthy if the network or user is not immediately ready for the message.
7. Some time later, these messages are received by the user, system device (printer), or another system.
8. Finally, the output message is purged from the system.

The following diagram shows the components of response time for typical IMS transactions as the previous section describes. The RTA data collector also supports miscellaneous variations of these components.

```
@------------------
                    !                  |
 1 Comm.   2 Comm.  3 Pgm.   4 Appl.   |5 Appl.    6 Output   7 Output  8 Output
  Msg.      Msg.     Msg.     Deqs     | Enqs.      Msg.       Recd.     Msg.
  Arrival   Queued   Queued   Msgs(GU) | Output     Dequeued   By User   Purged
                                       | (ISRT)
|---------+--------+--------+----------+----------+----------+---------+--------|
          |                 |          |
          |<--------------->|----------`
         Input Queue Time (I)

                   |<------>|
          Program Input Queue Time (PI)

                            |<-------->|
                        Processing Time (P)

          |<---------------------------->|
        Response Time 0 (R0) (Application)
                 |<------------------->|

                                        |<--------->|
                                     Output Queue Time (O)

          |<---------------------------------------->|
                      Response Time 1 (R1) (Host)
             |<--------------------------------->|
```

Figure 15. Components of IMS response time

RTA response time components

Input queue time (I)
: Time an input message that is sent by a communications input device (terminal) spent on the input queue before delivery to an application program. (From stage 2 - 4.) This time includes queue time, scheduling, program load, and program initialization.

Program input queue time (PI)
: Time an input message that is sent by an internal application program (program to program switch) spent on the input queue before delivery to an application program. (From stage 3 - 4.) This time includes queue time, scheduling, program load, and program initialization.

Processing time (P)
: Time transaction was processed by an application program until the first output message was prepared. (From stage 4 - 5.) This time includes message queue get unique, program execution, output message insert, wait for sync point, and program termination.

Response time 0 (R0)
: Internal systems application response time for the transaction. It is the sum of I (or PI) and P. (From stage 2 - 5.)

Output queue time (O)
: Time that an output message spends on the message queue before the message is dequeued for delivery to a system destination. (From stage 5 - 6.) This time includes output message insert, program termination, and some output message processing.

Response time 1 (R1)
: Total host response time for the transaction. It is the sum of Input Queue Time (or Program Input Queue Time), Processing Time, and Output Queue Time. Response Time 1 is greater than Response Time 0. However, R1 response times are *not* recorded for a transaction if it does not have an Output Queue Time component. Transactions with express output messages (messages that are issued before Sync Point) can have a Response Time 1 value, which is less than the Response Time 0 value, because such transactions issue a response before transaction completion.

  R1 times are not produced for non-recoverable inquiry transactions because the LTERM ID of the originating terminal is not available. (From stage 2 to 6.)

  Non-recoverable transactions are defined as INQUIRY=(YES,NORECOV) on the TRANSACT macro.

In addition to average response time, a count of transactions that are processed by IMS for that workload is displayed. The count of transactions indicates the significance of the average response time values. If the count is small, a high average response-time is insignificant. A count is given for each response-time component.

The rate is the number of transactions that are processed per minute. It is calculated by dividing the number of transactions that are processed by the elapsed time for the EPILOG interval. The rate increases as the workload on your IMS system increases and as your system is tuned.

Degradation data

The DETAIL display lists competing states with their associated times. This bottleneck data is also known as degradation data.

A transaction is in a competing state when it is being processed or is eligible to be processed but waiting. That is, it is either using processor, doing I/O, or waiting in line behind other transactions for its turn to do one of these tasks.

A transaction can wait in the message queue—for example, if the only MPR servicing its class is busy; or it can wait in a dependent region—for example, if it is queued for an IMS latch.

The competing states are categorized as follows.

- Using the CPU
- Scheduling waits
- Database I/O waits
- z/OS waits
- IMS waits
- Output waits
- External subsystem waits

The substates of the major types are indented, and their associated time and percentage figures are shown in parentheses.

Percentages and graphs: The percentage figures and the graph in the DETAIL report are a profile of how often during the sampling period the system was found in various competing states that occur during response time 0.

For example, according to Figure 13 on page 25, the system was:

- Using the CPU 18% of the time.
- Using the CPU in APPL 7.9% of the time.
- Using the CPU in IMS 10.1% of the time.

This display only shows competing states with associated percentage figures of 3% or more. The default is 5%; however, you can change this threshold figure by using the SET PLOTMIN command.

The graph arrows represent the percentage figures. The arrows in the graph are divided into 30% ranges. The first range is indicated by a hyphen (-), the second by an equal sign (=), and the third by a right-directed caret (>).

Time figures: The time figures in the TIME column of the degradation data are calculated by using their wait percentage and the logically associated response time component. For example, Wait for MPP is logically associated with the Input Queue Time response time component, because Wait for MPP occurs during Input Queue Time. Therefore, the wait time that is calculated for Wait for MPP is determined by multiplying its wait percentage by the time for Input Queue Time.

These calculations are completed differently depending on how many types of wait reasons there are under each execution state. For example, suppose that the system was waiting for GU in addition to MPP. Because Wait for GU is also logically associated with Input Queue Time, part of Input Queue Time must be associated with Wait for GU. Therefore, before the calculation is completed, the percentage that is associated with Wait for GU and Wait for MPP is adjusted so both wait reasons account for 100% of the Input Queue Time. This adjustment is done only for the calculation of the wait time and does not affect the actual display of the wait percentage.

Also, if there is a Program Input Queue Time, this time are added to Input Queue Time before the calculations are completed. The times are weighted based on the number of transactions that are associated with Input Queue Time and Program Input Queue Time.

Typically, the person responsible for system tuning is more concerned about the time figures that display than the percentage-derived times.

DETAIL display in a DBCTL environment

In a DBCTL environment, the DETAIL display is different from the one in the DETAIL report. Because there are no transactions in a DBCTL environment, response time data is not valid.

Figure 16 on page 29 shows an example of using the DETAIL keyword in a DBCTL environment.

```
+==============================================================================+
| Group = 3                     Symbolic Name = PAYROLL                          |
| Period:  19:30 to 19:45 on 08/29/15                 Elap = 15:00 M   I220 |
+------------------------------------------------------------------------------+
|                            DEGRADATION DATA                                  |
+------------------------------------------------------------------------------+
|Competing_State____________Wait_%_|0___1___2___3___4___5___6___7___8___9___0|
|Database I/O Waits            54.7 |------------=========>  .   .   .   .   .|
| DI21PART                     (9.4)|--->.   .   .   .   .   .   .   .   .   .|
| DI21PARO                    (41.5)|------------====>   .   .   .   .   .   .|
|IMS Waits                      7.5 |--->.   .   .   .   .   .   .   .   .   .|
| IRLM Conflict Wait           (5.7)|--> .   .   .   .   .   .   .   .   .   .|
|------------------------------------------------------------------------------|
+==============================================================================+
```

Figure 16. DETAIL report in a DBCTL environment

Summary workload (SUMMARY keyword)

The SUMMARY workload display is related to the DETAIL report in that it also shows both response time and bottleneck data. However, SUMMARY displays only a single line of output for each interval instead of giving a detailed breakdown of bottleneck states.

Each line contains the average transaction response time for that interval and the highest (worst) wait reason. You can browse through a concise summary of an entire day on one page or a number of days if the time interval is kept short enough and a range of days is specified.

The example that is shown in Figure 17 on page 30 might result from the following command:

```
DISPLAY SUMMARY GRP(3) STIME(10) ETIME(12) SDATE(08/29/15)
EDATE(08/29/15)
```

The same date that is specified in SDATE (StartDATE) and EDATE (EndDATE) limits the display to one day. GRP(3) is the shortened form for transaction group 3. Each line represents a single (RMF) interval, including the main wait reason and the average transaction response time. The plot graphically shows the total response that is broken into the main reason (plotted with *) for that interval and all the other reasons (-).

```
+-------------------------------------------------------------------------------+
|                                                                               |
| Transaction group =   3                                                       |
.************************************************
| Periods:      From 08/29/15 at 10:00 to 08/29/15 at 12:00                     |
| Misc Options: SUMMARY SINGLE                                                  |
|                                                                               |
|_______________________________________________________________________________|
|                         COMPETING TRANSACTIONS
+===============================================================================+
| PERIOD: 08/29/15 10:00   TO   08/29/15 12:00
+===============================================================================+
|DATE__START_END___MAIN_REASON(*) TIME(-)|0S___0.2____0.4____0.6____0.8_____10|
|08/29 10:00 10:15  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      10:15 10:30  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
|      10:30 10:45  DB PMFG021     7.06 S|*****************-------->  .      .|
|      10:45 11:00  DB PMFG021     9.85 S|*********************------------->.|
|      11:00 11:15  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
|      11:15 11:30  DB PMFG021     5.38 S|*************------> .      .      .|
|      11:30 11:45  DB PACCT03     1.73 S|***--->.      .      .      .      .|
|      11:45 12:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|-----------------------------------                                           |
|AVERAGE (RESP TIME 0)                                                         |
+==============================================================================
```

Figure 17. SUMMARY report

This report shows IMS response time for each 15-minute interval with main reason. For example, for the 10:45–11:00 period, average response time was 9.85 seconds. On further investigation, we find that the response time was slow in three areas and that the same database was the main reason all three times. This condition might result from excessive access or from the database that resides on a busy device.

The average response time during each period is indicated by hyphens (**-->**) that provide a quick graphic aid. The asterisks show how *much* of the response time was taken up by DB PMFG021, about 5.9 seconds.

SUMMARY display in a DBCTL environment

In a DBCTL environment, the SUMMARY display is different from the one in a SUMMARY report. Because there are no transactions in a DBCTL environment, the SUMMARY display shows wait percentages rather than response time data.

Figure 18 on page 31 shows an example of using the SUMMARY keyword in a DBCTL environment.

```
+==============================================================================+
| Group = 3                      Symbolic Name = PAYROLL                       |
| Period:  10:00 to 12:00 on 08/29/15                                   I220 |
+------------------------------------------------------------------------------+
|                           COMPETING STATES                                   |
+------------------------------------------------------------------------------+
|DATE__START_END___MAIN_REASON____________%_|0__1__2__3__4__5__6__7__8__9__0|
|08/29 10:00 10:15  USING CPU/APPL        36.8|---------==>.  .  .  .  .  .  .|
|      10:15 10:30  DB PMFG021            50.0|---------======>  .  .  .  .  .|
|      10:30 10:45  DB PMFG024            45.2|---------====> .  .  .  .  .  .|
|      10:45 11:00  DB PMFG021            39.0|---------===>  .  .  .  .  .  .|
|      11:00 11:15  DB PMFG021            51.1|---------======>  .  .  .  .  .|
|      11:15 11:30  USING CPU/APPL        37.6|---------===>  .  .  .  .  .  .|
|      11:30 11:45  USING CPU/APPL        35.9|---------===>  .  .  .  .  .  .|
|      11:45 12:00  DB PMFG024            48.1|---------======>  .  .  .  .  .|
|------------------------------------------------------------------------------|
+==============================================================================+
```

Figure 18. SUMMARY report in a DBCTL environment

Transaction volume (TRANSCOUNT keyword)

With the TRANSCOUNT keyword, you can request a different type of SUMMARY display, which shows the transaction volume in addition to the average response time. When you use TRANSCOUNT, EPILOG uses the short form of the wait reason description. For a description of these short form wait reasons, see Chapter 5, “Selecting EPILOG displays based on bottleneck factors,” on page 55.

The following figure shows a SUMMARY report that is generated by a typical SUMMARY report command.

```
CMD==> DISPLAY SUMMARY (TRANS) GRP(3) STIME(05) SDATE(08/29/15) EDATE (08/29/15)
 ******************************************************************************
 +==============================================================================+
 | TRANSACTION GROUP =   3                                                      |
 | PERIOD:  05:51 TO 13:15 ON 08/29/15                                          |
 +------------------------------------------------------------------------------+
 |                        COMPETING TRANSACTIONS                                |
 +------------------------------------------------------------------------------+
 |DATE__START_END___WAIT(*)__TRANS_TIME(-)|0S_____7_____14_____21_____28_____35|
 |08/29 05:51 06:00    WMP      60 32.24 S|*******************------------>   .|
 |      06:00 06:15    ISC     469  8.01 S|**----->      .      .      .      .|
 |      06:15 06:30    APL     452 18.50 S|**********--> .      .      .      .|
 |      06:30 06:45    APL     390 15.27 S|*******---->  .      .      .      .|
 |      06:45 07:00    PSB     464  7.01 S|****-->.      .      .      .      .|
 |      07:00 07:15    DMB     531 13.50 S|*******--->   .      .      .      .|
 |      07:15 07:30    MFS     729 11.12 S|****-->.      .      .      .      .|
 |      07:30 07:45    QBL     720  2.76 S|**>    .      .      .      .      .|
 |      07:45 08:00    PIW     685  3.02 S|**>    .      .      .      .      .|
 |      08:00 08:15    SHM     734  3.06 S|**>    .      .      .      .      .|
 |      08:15 08:30    WMP     862  2.47 S|*>     .      .      .      .      .|
 |------------------------------------------------------------------------------|
 |AVERAGE (RESP TIME 0)            10.63 S                                      |
 +==============================================================================+
```

Figure 19. SUMMARY report with transaction counts

TRANS is the short form for TRANSCOUNT, which causes the transaction volume to display. GRP(3) is the short form for workload definition group 3. The same date that is specified in SDATE (StartDATE) and EDATE (EndDATE) limits the display to one day. Each line of the panel represents a single (RMF) interval, and includes the main wait reason and the average transaction response time. The plot graphically separates the total response into the main reason for that interval (plotted with *) and all the other reasons (plotted with -).

Transaction group summary (GSUM keyword)

The GSUM keyword displays the transaction rate and average response time by transaction group. Using the system navigator, the user can obtain detail degradation data or resource data for the selected transaction group.

The following list shows the operands for the GROUPSUMMARY (GSUM) keyword.

R0
: Response time 0 values

R1
: Response time 1 values

OQ
: Output queue values

IQ
: Input queue values)

PI
: Program input queue values

PR
: Processing values

The operands for the GSUM keyword are used to select which response time components are to be displayed. The default is the R0 value.

Figure 20 on page 32 shows an example of a detail GSUM display.

```
CMD==> DIS GSUM(R0) TDAY
*******************************************************************************
+=============================================================================+
| Group Summary                                                               |
| Period:  10:29 to 10:44 on 08/29/15                    Elap = 15:00 M IMST  |
+-----------------------------------------------------------------------------+
|                 RESPONSE TIME 0 VALUES |                                    |
|Group___Name______Rate___Count____Time__|0S___0.2____0.4____0.6____0.8_____10|
|      System      4.67      70    0.99 S|--->   .      .      .      .      .|
|      Other        .00       0    0.00 S|.      .      .      .      .      .|
|  01  Billing     0.07       1    1.49 S|-----> .      .      .      .      .|
|  02  Shipping    1.93      29    1.21 S|---->  .      .      .      .      .|
|  04  Account    11.28      53    1.00 S|-->    .      .      .      .      .|
|  05  Payroll     2.13      30    0.41 S|>      .      .      .      .      .|
|  06  Inventory   3.04      34    1.15 S|--->   .      .      .      .      .|
|  07  Orderproc   2.80      42    6.91 S|----------------------->    .      .|
+=============================================================================+
```

Figure 20. DETAIL GSUM display

Resource displays (Rccc keywords)

Resource displays are actually many different panels with many keywords, each with information that pertains to a particular area of performance. Some of this data is extracted directly from the information that is gathered by RMF, but most of it is generated by the collector.

You can use different operands to limit the scope of the resource display. For example, use RDEV to select by volume or unit; use ROSP and RVSP to select by subpool number; and use RPAG to select by region name.

Table 6 on page 32 summarizes the types of resource displays and the keywords that invoke them.

Note: NORESC disables these keywords. If you disable a keyword, its respective report is not available.

The syntax for NORESC is:

```
NORESC(ccc)
```

where *ccc* is the name of the resource collector (the last three characters of the Resource Display Rccc Keyword), except for the RDDN resource collector, which is RDN.

Table 6. Resource panel types

Keyword	Description	Operands
RALL	Display all resource panels	None

Keyword	Description	Operands
RBUF	IMS sequential buffering storage statistics	None
RCIP	IMS CIOP statistics	None
RCFP	IMS combined fixed pool statistics (CESS, CIOP, HIOP, LUMC, LUMP, EMHB, FPWP, SPAP, AOIP)	None
RCMP	IMS combined pool statistics (MAIN, DMBW, PSBW, EPCB)	None
RCPU	IMS CPU activity	None
RDAS	IMS data set statistics	None
RDBE	IMS database error statistics	Database name
RDDN	IMS ddname statistics	DDname
RDEV	IMS device statistics	Unit or volume
RDMP	IMS DMB pool statistics	None
RECA	IMS receive any pool statistics	None
RFPI	General fast path information	None
RFPO	Fast Path output thread statistics	None
RFPP	Fast Path buffer pool statistics	None
RFPR	Fast Path region statistics	Region name
RFRP	IMS FRE pool statistics	None
RINF	General IMS and z/OS system information	None
RLAT	IMS latch statistics	None
RLOG	IMS logging statistics	None
RMFP	IMS message format buffer pool statistics	None
ROSP	IMS ISAM/OSAM buffer pool statistics	Subpool number
RPAG	IMS page and storage information	Region name
RPIP	IMS PI enqueue pool statistics	None
RPSP	IMS PSB pool statistics	None
RQBP	IMS QBUF statistics	None
RSAP	IMS SAP pool statistics	None
RSCD	IMS schedule statistics	None
RSRM	IMS SRM statistics	Region name
RTXU	Unschedulable transaction statistics	Transaction name
RVSO	Fast Path VSO dataspace statistics	None
RVSP	IMS VSAM buffer pool statistics	Subpool number

Table 6. Resource panel types (continued)

To start these panels explicitly, use the appropriate keyword of the DISPLAY command. For example:

```
DISPLAY RCIP
DISPLAY RDAS
DISPLAY RLOG
```

You can also start resource panels automatically with the AUTOMATIC keyword on a DETAIL display. For more information, see "Automatic Analysis" on page 70.

Chapter 11, "Reference: Sample resource panels in EPILOG displays," on page 101 shows some sample output for the various resource panel displays.

Collection options (COPT keyword)

The COLLECTOPT keyword displays collection options for the interval that is displayed.

The short name for COLLECTOPT is COPT. There are no operands.

The following COPT display example shows how the system displays COPT results based on collection options, for example, collecting I/O statistics by database:

```
CMD==> DISPLAY COPT
********************************************************************************
+==============================================================================+
| PERIOD:  20:45 TO 20:59 ON 08/29/15                  ELAP =14:13 M     IM3D |
+------------------------------------------------------------------------------+
|                               COLLECTION OPTIONS                             |
+------------------------------------------------------------------------------+
| DEGRADATION CYCLE = 0.50 S                                                   |
| RMF SYNCHRONIZED                                                             |
| COLLECTING I/O STATISTICS BY DATABASE                                        |
+------------------------------------------------------------------------------+
|                               GROUP DEFINITIONS                              |
+------------------------------------------------------------------------------+
|GROUP # 1 TRAN  =  TRAN1    TRAN2    TRAN3    TRAN4    TRAN5    TRAN6         |
|                   TRAN7    TRAN8    TRAN9                                    |
|          PSB   =  JAN                                                         |
|          CLASS =     1                                                        |
|GROUP # 2 LTERM =  LTERM0   LTERM1   LTERM3   LTERM4   WTOR                     |
|GROUP #10 TRAN  =  TRANA    TRANB    TRANC    TRAND    TRANE                    |
|GROUP #15 TRAN  =  *AVE     JA******                                           |
|          CLASS =     1        2        3                                      |
+==============================================================================+
```

Figure 21. COPT display example

Workloads

Both the DETAIL and SUMMARY bottleneck reports require that you specify a workload. A workload is a way of identifying what you want to report on, such as transaction groups or system-wide bottleneck.

See the following table for a list of possible workload-type keywords.

Table 7. Workload keywords

Keyword	Short	Description	Operand
GROUP	GRP	Displays degradation for a transaction group or groups; or displays symbolic name if you define the group with SUMNAME.	1–30 groups, or a symbolic name
SYSTEM	SYS	Displays system-wide degradation.	None
OTHER	OTH	Displays bottlenecks for all the transactions that are not defined in any group.	None

The collector records bottleneck data for the following workload types:

- Transaction groups (by number or symname)

- System-wide (total of entire system)
- Other transactions

Do not mix different types of workloads in one request. For example, it is not valid to ask for information about transaction group 2 and the system at the same time.

Transaction groups

Transaction groups are user-defined groups in which certain types of transactions are accessed together. To display degradation for a specific transaction group, select it by name or number with the keyword GROUP (or the abbreviation GRP).

The following is an example of how to issue the DISPLAY DETAIL GROUP command.

```
DISPLAY  DETAIL GROUP(2)

DISPLAY  SUMMARY GRP(BILLING)
```

System-wide bottlenecks

To display bottlenecks for the system, simply specify the keyword SYSTEM.

The following is an example of how to issue the SYSTEM command.

```
DISPLAY  DETAIL SYSTEM

DIS DTL SYS
```

Executing transactions

The DETAIL and SUMMARY displays include information for all applicable competing transactions. To limit the displays to executing transactions only, use the EXECUTING keyword.
The following is an example of how to issue the **DISPLAY DETAIL EXECUTING** command.

```
DISPLAY DETAIL EXECUTING

DIS SUM EXC
```

Figure 22 on page 36 contains the report that is generated from the **DISPLAY DETAIL EXECUTING** command.

```
+==========================================================================+
| Transaction Group =   3        Symbolic Name = PAYROLL                   |
| Period:  19:30 to 19:45 on 08/29/15                  Elap = 15:00 M   I220 |
+--------------------------------------------------------------------------+
|                          RESPONSE TIME DATA                              |
+--------------------------------------------------------------------------+
|Response_Component_____Avg. Rsp. Time______Trans. Count_____Rate (per min.)  |
|Input Queue        |     0.00 S       |        0       |        .00        |
|Pgm Input Queue    |     0.39 S       |       25       |       4.10        |
|Processing         |     1.64 S       |       24       |       3.93        |
|Response time 0    |     2.05 S       |       24       |       3.93        |
|Output Queue       |     0.03 S       |       23       |       3.77        |
|Response time 1    |     0.00 S       |        0       |        .00        |
+--------------------------------------------------------------------------+
|                             DEGRADATION DATA                             |
+--------------------------------------------------------------------------+
|Executing_State_________Time_______|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU             0.05 S    1.4|.   .   .   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.05) S  (1.4)|.   .   .   .   .   .   .   .   .   .   .|
|z/OS Waits            0.59 S   18.3|------->.   .   .   .   .   .   .   .   .|
| BLDL I/O           (0.55) S (16.9)|------> .   .   .   .   .   .   .   .   .|
| Cross Memory Page  (0.05) S  (1.4)|.   .   .   .   .   .   .   .   .   .   .|
|IMS Waits             1.00 S   31.0|------------>   .   .   .   .   .   .   .|
| Sync Point Wait    (0.05) S  (1.4)|.   .   .   .   .   .   .   .   .   .   .|
| Iswitched to CTL   (0.96) S (29.6)|----------->.   .   .   .   .   .   .   .|
|Database I/O Waits    1.59 S   49.3|-----------=======>.   .   .   .   .   .|
|-----------------------------------                                       |
|Average (Resp Time 0) 2.05 S                                              |
+==========================================================================+
```

Figure 22. Displaying executing transactions

The Other keyword

Often, many transactions fall outside a convenient group or are unable to be grouped. The OTHER keyword displays information for all the transactions that are not grouped.

The following is an example of how to issue the OTHER command.

```
DISPLAY DETAIL OTHER

DISPLAY SUM OTH
```

Time specifications

Without time specifications, EPILOG tries to display the data you request for every interval in the EPILOG datastore, every time you press Enter. It is better to request data from a narrowly defined time range.

You can use the date/time selection keywords to tailor the displays to limit the amount of data to some period or periods. In general, supply a starting date and time, and ending date and time:

```
DIS GRP(2) SDATE(05/16/15) STIME(0800) EDATE(05/19/15) ETIME(1600)
```

This command displays periods, which begin after the start time and before the end time on each day selected. This display is the BAND display. In Figure 23 on page 37, the shaded areas represent the intervals that are selected for monitoring—the 8-hour period of 0800 through 1600 on 05/16, 05/17, 05/18, and 05/19, a total of 32 hours.

```
0     12    0     12    0     12    0     12    0

                    |   |/////|  |  |/////|  |  |/////|  |  |/////|   |
                    |    /////   |   /////   |   /////   |   /////    |
                    |   |/////|  |  |/////|  |  |/////|  |  |/////|   |
                    |    /////   |   /////   |   /////   |   /////    |
                    |   |/////|  |  |/////|  |  |/////|  |  |/////|   |
                    |    /////   |   /////   |   /////   |   /////    |
                    |   |/////|  |  |/////|  |  |/////|  |  |/////|   |
                    |    /////   |   /////   |   /////   |   /////    |
                    |   |/////|  |  |/////|  |  |/////|  |  |/////|   |

                      05/16        05/17       05/18        05/19

                  SDATE(05/16) STIME(0800) EDATE(05/19) ETIME(1600) BAND
```

Figure 23. Time band specification

Alternately, you can specify the RANGE keyword, which displays data for the entire period, not just for part of each day. In Figure 24 on page 37, the shaded area represents the interval that is selected to monitor, from 0800 on 05/16 through 1600 on 05/19, or 80 hours. BAND tends to be more useful than RANGE when you analyze performance data. BAND is the default and you can leave it out.

```
0     12    0    12    0      12     0    12    0
                @----------------------------------------------------------
                |       |/////////|/////////|/////////////|//////|       |
                |        /////////|/////////|/////////////|//////        |
                |       |/////////|/////////|/////////////|//////|       |
                |        /////////|/////////|/////////////|//////        |
                |       |/////////|/////////|/////////////|//////|       |
                |        /////////|/////////|/////////////|//////        |
                |       |/////////|/////////|/////////////|//////|       |
                |        /////////|/////////|/////////////|//////        |
                |       |/////////|/////////|/////////////|//////|       |
                %----------------------------------------------------------`
                        05/16       05/17       05/18       05/19

                 SDATE(05/16) STIME(0800) EDATE(05/19) ETIME(1600) RANGE
```

Figure 24. Time range specification

Specify explicit start and end dates and times in the following formats:

```
SDATE(date) and EDATE(date)
```

where *date* can be:

Julian:
yyddd (ddd = 001 through 366)

Gregorian:
mmddyy

mm/dd/yy

mm/dd (assumes current year)

(assumes current month, year)

```
STIME(time) and ETIME(time)
```

where *time* can be:

h
The hour without leading 0

hh
Two-digit hour or leading 0

hmm
Hours and minutes without leading 0

hhmm
Hours and minutes with leading 0 or two-digit hour

Enter time specifications using the 24-hour clock (not using A.M. or P.M.). Count the hours after midnight, that is, from 1 to 24. For example, 12:00 noon is 1200, 1:00 p.m. is 1300, 2:00 p.m. is 1400, and 3:25 p.m. is 1525. Table 8 on page 38 lists all the various time specification keywords:

Table 8. Date/Time keywords

Keyword	Short	Description	Operand
BAND		Display data between the start and end times on the specified days; BAND is the default.	
DAYOFWK	DAY	Days of the week. Possible operands: • M MON MONDAY • TU TUE TUESDAY • W WED WEDNESDAY • TH THU THURSDAY • F FRI FRIDAY • SA SAT SATURDAY • SU SUN SUNDAY • ALL (default) • WKDAY WEEKDAY • WKEND WEEKEND	1–9 chars
ENDDATE	EDATE	Ending date	Date
ENDTIME	ETIME	Ending time	Time
EXDATE	EXD	Exclude the indicated dates from the display. (Only valid with DISPLAY and SET commands.)	Dates
LASTMONTH	LMN	Start the first day and end the last day of the previous month.	
LASTWEEK	LWK	Start Monday of the previous week and end seven days later.	
NUMDAY		Specifies a range beginning at the start of data collection up to NUMDAYs before today. Only valid with EXCLUDE and PURGE commands. For example, if NUMDAY=30, that means PURGE or EXCLUDE all data up to 30 days before today.	1–32767
RANGE		Display data from the start time and date to the end time and date. (BAND is the default.)	
STARTDATE	SDATE	Starting date	Date
STARTTIME	STIME	Starting time	Time
THISMONTH	TMN	Start the first day and end the last day of the current month.	
THISWEEK	TWK	Start Monday of the current week and end seven days later.	
TODAY	TDAY	Start and end on today’s date.	
YESTERDAY	YDAY	Start and end on yesterday’s date.	

Following are some common examples of different types of date and time specifications for transaction group 2:

Display data from 08:00 to 17:00 yesterday.

```
DIS GRP(2) STIME(8) ETIME(17) YDAY
```

Display data every day this week from 18:00 to midnight. (Note that because you omitted ETIME, 24:00 is assumed.)

```
DIS GRP(2) STIME(1800) THISWEEK
```

Display all data from 8:00 to 12:00 on 6/12 through 6/18.

```
DIS GRP(2) STIME(8) ETIME(12) SDATE(6/12) EDATE(6/18)
```

Display all data from 8:00 on 6/12 through 12:00 on 6/18.

```
DIS GRP(2) STIME(8) ETIME(12) SDATE(6/12) EDATE(6/18) RANGE
```

Display all data from Monday of last week between 9:00 and 12:00.

```
DIS GRP(2) STIME(9) ETIME(12) LWK DAY(MONDAY)
```

IMS ID selection

The design of EPILOG requires that data is collected into an EPILOG datastore (EDS) by only one system at any one time. Normally this means that all records within your EPILOG datastore are stamped with the same IMS ID.

However, there is nothing wrong with having records with different IMS IDs in the same datastore; this happens whenever you changed your system's IMS ID.

By default EPILOG ignores the IMS ID stamped within a record and treats all the records within the same EPILOG datastore as if they originated from the same system.

On occasion, you might instruct EPILOG to display only records with a certain IMS ID. Add the SYSID keyword to any display request:

```
DISPLAY GRP(2) SYSID(A083)
```

Keyword default setting (SET command)

If you use the DISPLAY command, you can often find that you enter the same set of options for each command such as start/end date and time. To reduce the amount of typing, EPILOG provides a command that is called SET, which establishes defaults for some types of command keywords.

To set defaults for the DISPLAY command, use the following syntax:

```
SET keywords
```

Where the *keywords* specify the defaults you want to set.

For example, if you intend to issue a series of commands that are targeted at yesterday (for this example, yesterday is considered to be 08/29/15) from 9:00 AM to 5:00 PM, you might issue the SET command:

```
SET YESTERDAY STIME(9) ETIME(17)
```

before you made any other entries. The result is illustrated in Figure 25 on page 40.

```
---------------------------------------------------------------------------
|                                                                           |
| Periods:      Starting on 08/29/15 at 9:00 and Ending on 08/29/15 at 17:00  |
| Misc Options: PLOTMIN(5) LIMIT(10)                                        |
|                                                                           |
 ---------------------------------------------------------------------------
```

Figure 25. SET default options

For the rest of your session, any time you issue a DISPLAY command, EPILOG assumes that you entered the YESTERDAY STIME(9) ETIME(17). A request such as:

```
DIS GRP(2) SUMMARY YDAY STIME(9) ETIME(17) COMBINE(1H)
```

Can now be abbreviated as:

```
DIS GRP(2) SUM COMB(1H)
```

The result is shown in Figure 26 on page 40.

```
+------------------------------------------------------------------------------+
| TRANSACTION GROUP =   2                                                      |
| PERIOD: 08/29/15 09:00   TO   08/29/15 17:00                                 |
|------------------------------------------------------------------------------|
|COMPETING TRANSACTIONS
+==============================================================================+
|DATE__START_END___MAIN_REASON(*)_TIME(-)|0S___0.2___0.4___0.6___0.8____10|
|08/29 09:00 10:00  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      10:00 11:00  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
|      11:00 12:00  DB PMFG021     7.06 S|*****************-------->  .      .|
|      12:00 13:00  DB PMFG021     9.85 S|*******************-------------->.|
|      13:00 14:00  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
|      14:00 15:00  DB PMFG021     5.38 S|*************------> .      .      .|
|      15:00 16:00  DB PACCT03     1.73 S|***--->.      .      .      .      .|
|      16:00 17:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|-----------------------------------                                           |
|Average (Resp Time 0)                                                         |
+==============================================================================+
```

Figure 26. Display using SET default options

You can view the current default settings at any time by entering the SET command without any operands. Adding the keyword CLEAR clears all default settings (as if EPILOG stopped and restarted).

Only some of the valid DISPLAY keywords can be SET; these break into three categories:

- All date and time specification keywords (STIME, BAND, and THISMONTH). (See "Time specifications" on page 36.)
- Exception limit keywords. (See Chapter 5, "Selecting EPILOG displays based on bottleneck factors," on page 55.)
- Some of the miscellaneous options. (See Table 9 on page 40.)

Table 9 on page 40 outlines the options, which can be used with SET.

Table 9. Option keywords for SET

Keyword	Short	Description	Operand
AMATRIX		Specifies an automatic-analysis matrix to force the display of certain resource panels: applies only to DETAIL reports.	1–7 characters
AUTOMATIC	AUTO	Displays the resource panels that are indicated by AMATRIX for the highest wait reason; applies only to DETAIL and SUMMARY reports.	

Keyword	Short	Description	Operand
Table 9. Option keywords for SET (continued)			
LIMIT	LIM	Limits the number of records that match the selection criteria. Zero implies no limit (default = 10 for batch, 0 for fullscreen).	0–32767
LOGON		Turns on screen logging to the EILOG data set.	
LOGOFF		Turns off screen logging to the EILOG data set.	
MANUAL	MAN	No automatic resource displays (default: applies only to a DISPLAY DETAIL request).	
PLOTMIN	PMIN	Plot percentage threshold; applies only to DETAIL displays and default = 5.	0–99
PLOTPCT	PPCT	Used with SUMMARY and ON keywords; causes the plotting of degradation reason percentage rather than response time.	
PLOTTIME	PTME	Used with SUMMARY and ON keywords; causes the plotting of bottleneck reason time.	
SHORT		Shows a short format title block.	
SYSID		Displays data from the system only if it is marked by the indicated IMS ID.	Four-character IMS ID
TITLE		Supplies a single-spaced report title line.	1–72 character report heading
TITLE2		Supplies a double-spaced report title line.	1–72 character report heading
TITLE3		Supplies a triple-spaced report title line.	1–72 character report heading
TOTAL	TOT	Used with COMBINEd SUMMARY and DETAIL keywords; causes the total transaction times to be displayed (rather than AVG, which is the default).	

Full-screen control (CONTROL command)

In full-screen mode, EPILOG can use the extended color capabilities of 3279-type terminals, which support this option. Extended color support is turned on automatically. If your terminal is not configured to support it, EPILOG reverts to high- and low-intensity highlighting. You can control extended color support with the CONTROL COLOR (also known as CNTL COLOR) command and its suboperands.

OFF
Turns off extended color mode.

None
No operand indicates that color mode is turned on and the current color definition is used.

”string”
A 1- to 8-byte string sets the color areas for 3279 operation and turns color mode on. Each position is a color designation (or a period to indicate no change) for the specific color areas.

CAUTION: Errors can result if you turn extended color mode on (using either CNTL COLOR or CNTL COLOR string) while you use a non-color terminal. (PROG471 and IKT00405I SCREEN ERASURE CAUSED BY ERROR RECOVERY PROCEDURE are two likely error messages.) If this error occurs accidentally, use the PA1 key to reset EPILOG into non-color mode (see the *IBM OMEGAMON for IMS on z/OS User's Guide*.)

The default color assignments for the eight-character string are shown in Table 10 on page 42:

Table 10. Default color assignments

Area	**Color**	**Letter**
Headings and separator lines	Green	G
Warnings and errors	Red	R
Data entry line	White	W
Odd display panels	Turquoise	T
Even display panels	Yellow	Y
Highlighted areas	Red	R
Measure line	Blue	B
Title blocks and graph lines	White	W

The highlighting options are shown in Table 11 on page 42:

Table 11. Attribute options

Attribute	**Letter**
Blink	B
Reverse	B
Underscore	U
Default	D

Figure 27 on page 42 shows the screen that displays when you enter the CNTL COLOR command.

```
Control color   'G R W T Y R B W '
Control hilite  '. . . . . . . . '
                 | | | | | | | |
                 | | | | | | | | Title blocks and graph lines
                 | | | | | | Measure line
                 | | | | | Highlighted areas
                 | | | | Even display panels
                 | | | Odd display panels
                 | | Data entry line
                 | Warnings and errors
                 Headings and separator lines

                 Colors:
                    B = Blue   G = Green      P = Pink
                    R = Red    T = Turquoise  W = White
                    Y = Yellow

                 Attributes:
                    B = Blink       R = Reverse
                    U = Underscore  D = Default

                 A period (.) signifies no change
```

Figure 27. Default color assignments

To change the color of a particular area, type the letter of a color corresponding to the letter that points to the area. Then, press Enter.

You can also assign certain attributes to an area. Such attributes can make any text that appears there blink, appear in reverse (dark letters on light background), or be underscored. To change the attributes of an area, type the letter of the attribute you want on top of the period, which points to the area.

```
Control color   'G R W T Y G B W '
Control hilite  '. . R . . . . . '
                 | | | | | | | |
                 | | | | | | | Title blocks and graph lines
                 | | | | | | Measure line
                 | | | | | Highlighted areas
                 | | | | Even display panels
                 | | | Odd display panels
                 | | Data entry line
                 | Warnings and errors
                 Headings and separator lines
```

Figure 28. Color and attribute modification

In this example, “Highlighted areas” are changed to green, and the data entry line is reversed.

Table 12 on page 43 shows all the options that you can use with the CONTROL command.

Table 12. CONTROL keywords

Keyword	Operands	Description
COLOR	OFF color mask	Turns extended color support off. Turns extended color support on and sets the 8-byte color mask.
FOLDOFF		CNTL LOG does not fold output to uppercase.
FOLDON		CNTL LOG folds output to uppercase
HILITE	Highlight mask	Turns extended color support on and sets the 8-byte highlight mask.
HOLD		Changes the display mode; holds the topmost display on the screen.
LOG		Logs the current display to the EILOG data set. (PF9/PF21)
MODE		Advances the display mode (from PAGE to ROLL to HOLD to ONE and back to PAGE). (PF2/PF14)
ONE		Changes the display mode; allows only one display on the screen at a time.
PAGE		Changes the display mode; allows as many displays on the screen as can be accommodated.
RECALL		Recalls the previous command to the input line. (PF4/PF16)
ROLL		Changes the display mode; discards the topmost display each time you press Enter and rolls up the displays beneath it.
SCROLL	UP *n* DOWN *n*	Scrolls up *n* displays or screens. (PF7/PF19) Scrolls down *n* displays or screens. (PF8/PF20) *n* is optional; if you omitted it, EPILOG assumes 1. If **M** is entered, the display scrolls up or down the maximum value.
TITLE		Toggles the title block display on or off. (PF5/PF17)

Reporting in a multiple data store environment

You can use multiple datastore reporting to access multiple EPILOG data stores (EDSes) in a single EPILOG reporter session. You can also dynamically allocate and free EDS without exiting a session.

Automatically allocating an EDS

To automatically allocate an EDS when a reporter session begins, add a datastore name to the KEIEDS member in the RKANPAR data set. EPILOG reads this member at reporter initialization and allocates and opens the EDS specified.

Because the reporter uses only one EDS at a time, you can specify only one EDS in KEIEDS. However, you can include as many comments as you want. Enter the comments either by preceding them with asterisks in column 1 or by entering them after the datastore name. The datastore name itself can display anywhere in a line, provided it is only preceded by blanks.

To deallocate or allocate another datastore, use the DATASTOR command, as described in "Dynamically allocating and deallocating datastores (DATASTOR command)" on page 44.

Dynamically allocating and deallocating datastores (DATASTOR command)

You can allocate and free datastores dynamically during a reporter session by using the DATASTOR command. Changes that you make with this command are effective for the life of the current reporter session only.

The syntax for maintaining the active datastore list is as follows:

```
DATASTOR  USE  EDS(dsname)
```

or

```
DATASTOR  DROP
```

USE

Makes the specified datastore available for use by the reporter. Any datastore that is previously allocated becomes unavailable.

This command automatically verifies that the newly allocated file is a valid EDS. An EDS is valid only if it meets all of the following criteria:

- It is a VSAM KSDS.
- It has an LRECL less than 32K.
- Its key length is 32.

DROP

Makes the datastore unavailable for use by the reporter.

Each time a DATASTOR command is processed, the INQUIRE SUMMARY display is automatically started, providing statistics on the EDS that is active. See "Displaying EDS status (INQUIRE command)" on page 45 for more information about the INQUIRE command.

While the reporter is acquiring data from the EDS, the following display (Figure 29 on page 45) displays approximately every 30 seconds. You can stop this display at any time by pressing the Attention key.

```
*******************************************************************************
*                                                                             *
* +-----------------------------------------------------------------------+  *
* | DISPLAY RALL COMBINE                                                  |  *
* +-----------------------------------------------------------------------+  *
*                                                                             *
*  The delay in satisfying the current request is due to the amount of data   *
*   retrieval necessary to process it or possible system contention/delays    *
*                                                                             *
*                                                                             *
*  Elapsed time of request     = 00:31 (MM:SS)                                *
*                                                                             *
*  Number of records processed = 200                                          *
*                                                                             *
*******************************************************************************
```

Figure 29. Command execution delay display

Displaying EDS status (INQUIRE command)

You can display information about the status of the EPILOG datastore and its contents with the EPILOG INQUIRE command.

Workload type, resources, TITLE, and SUMMARY are the only allowable keywords.

Detail datastore statistics

If you do not enter keywords with the INQUIRE command, the following information displays for the available EPILOG data store. The information that is shown in Figure 30 on page 45 is displayed for the available EDS.

```
                              Datastore Statistics

   For:          CANDLE.EP.SAMPLE.EDSB
   Utilization:  87%

   Allocated Control Intervals -     1125
   Free Control Intervals      -       88
   Number of data records      -    24237
   Number of Extents           -        1
   Control Interval Size       -     8192
   Dataset LRECL               -    32700
   High allocated RBA          -  9216000

   Date and Time Range:   08/29/15 03:15 - 08/30/15 23:45
```

Figure 30. INQUIRE detail display

INQUIRE (or INQ) shows the date and time of the first and last record in the EDS for the workload, resource, or other display type. INQUIRE (or INQ) also displays the number of data records and their total length in bytes.

Summary datastore statistics

The SUMMARY keyword displays a summary status report for the available EPILOG datastore (EDS). An example of the display that is produced by the INQUIRE SUMMARY command is shown in Figure 31 on page 46.

```
                    Summary Datastore Statistics

        Date and Time Range                Utilization   Xt
        -------------------------------    -----------   --
    1  CANDLE.EP.SAMPLE.EDSB
        08/29/15 03:15 - 08/30/15 23:45            87%    1



             Enter “d” to select detailed datastore statistics
```

Figure 31. INQUIRE summary display

The display shows the creation date and time, the date, and time the EDS last updated, the percent of usage, and the number of extents. You can go to the detail display shown in Figure 30 on page 45 by entering D next to the datastore name.

The INQUIRE SUMMARY display is automatically started at the beginning of a reporter session and any time a datastore is allocated or deallocated during a session.

Datastore statistics by group

You can use the GROUP, or GRP, keyword to display the status of the EDS for a certain group. For instance, the command `INQ GRP(2)` shows you how much space transaction group 2 records are using, as shown in Figure 32 on page 46.

```
-----------------------------------------------------------------------------
|                                                                             |
| Workloads:    Group - 2                                                     |
| Data Store:   EPILOG.DATA.STORE.IMS                                         |
 -----------------------------------------------------------------------------

.************************************************
        First Record Written on - 08/29/15  at  02:15
        Last Record Written on  - 08/29/15  at  14:45
        Number of Records -     1901
        Length of 1901 data records = 642991
```

Figure 32. EPILOG INQUIRE GRP(2) command

Specifying index and data buffers (REOPEN command)

By default, the EPILOG reporter opens the EDS with three data buffers and two index buffers. There are circumstances in which the reporter can operate more efficiently if the number of VSAM data and index buffers are increased.

The number of VSAM data and index buffers can be increased by using the REOPEN command, which accepts the following keywords:

BUFND
: Number of data buffers

BUFNI
: Number of index buffers

For example, the command

```
REOPEN BUFND(5) BUFNI(7)
```

closes and reopens five data buffers and seven index buffers in the VSAM cluster by using the buffer specifications that you request.

After the VSAM cluster is reopened, the INQUIRE SUMMARY display (shown in INQUIRE summary display) appears.

Report separators (PAGESEP command)

When you are using EPILOG as a batch reporter, you may have special display requirements, such as titles, page separators, and so forth. You can use the EPILOG PAGESEP command to provide visual separators between sets of reports on the RKEIOUTR data set. You can create either report separators or routing distribution headers. The titles are boxed with asterisks.

The PAGESEP command uses the following keywords:

BLOCK
Block-letter title line

DUPLICATE/DUP
Number of separator pages to print

TITLE
Single-spaced title line

TITLE2
Double-spaced title line

TITLE3
Triple-spaced title line

For example:

```
PAGESEP DUP(3) BLOCK('ROOM 1') TITLE3('SPECIAL REPORT') -
  TITLE('TO THE DIRECTOR')
```

You can use TITLE, TITLE2, TITLE3, and BLOCK together for one command.

As shown in the following example, the PAGESEP command produces the page separator three times. This output is placed on the RKEIOUTR data set.

```
EPILOG DEGRADATION REPORT                             08/29/15  9.27.38 PAGE   1
COPYRIGHT (C) 1995 CANDLE CORPORATION


********************************************************************************
*                                                                              *
*                                                                              *
*                     ****   ***   ***  *   *         *                        *
*                     *   * *   * *   * ** **        **                        *
*                     *   * *   * *   * * * *       * *                        *
*                     ****  *   * *   * *   *         *                        *
*                     * *   *   * *   * *   *         *                        *
*                     *  *  *   * *   * *   *         *                        *
*                     *   *  ***   ***  *   *       *****                      *
*                                                                              *
*                                                                              *
*                               SPECIAL REPORT                                 *
*                              TO THE DIRECTOR                                 *
*                                                                              *
*                                                                              *
********************************************************************************
```

Figure 33. PAGESEP output

Chapter 4. Combining elements for the EPILOG reporter

You can create summary and detail displays for single and multiple intervals.

RMF single and multiple interval displays

The RMF interval is typically used on a z/OS system. The length of this interval can range from 15 minutes to 1 hour from installation to installation. The degradation and resource records that EPILOG collects are typically in sync with RMF to make it easier to compare performance data.

As convenient as this type of synchronization is, the RMF interval is not always the best unit of measurement to use to obtain performance data. RMF data that is broken down to the RMF interval level can provide more detail than you really want.

To give more reporting versatility, EPILOG shows degradation and resource displays either in an RMF single-interval state or combined into multi-intervals. The keywords SINGLE and COMBINE are mutually exclusive; SINGLE is the default. COMBINED records are kept in memory and are not part of the EDS.

Single-interval summary display

You can issue the following command if you want to generate a SUMMARY report, for example, for a two-hour time span on a number of days.

```
DISPLAY SUMMARY SYS SDATE(08/29/15) STIME(10) EDATE(08/31/15) ETIME(12)
```

The command generates the report that is shown in Figure 34 on page 49.

```
-----------------------------------------------------------------------------
|                                                                             |
| Workloads:    SYSTEMWIDE                                                    |
| Periods:      Starting on 08/29/15 and Ending on 08/31/15                   |
| Misc Options: SUMMARY SINGLE                                                |
|                                                                             |
 -----------------------------------------------------------------------------
+=============================================================================+
| SYSTEMWIDE                                                                  |
| Period: 10:00 to 12:00 on 08/29/15 to 08/31/15                              |
+-----------------------------------------------------------------------------+
+=============================================================================+
|DATE__START_END___MAIN_REASON__TIME___|0S_____2______4______6______8_____10|
|08/29 10:00 10:15  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      10:15 10:30  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
|      10:30 10:45  DB PMFG021     7.06 S|*****************-------->  .      .|
|      10:45 11:00  DB PMFG021     9.85 S|*********************------------->.|
|      11:00 11:15  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
|      11:15 11:30  DB PMFG021     5.38 S|*************------> .      .      .|
|      11:30 11:45  DB PACCT03     1.73 S|***--->.      .      .      .      .|
|      11:45 12:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|08/30 10:00 10:15  USING CPU/APPL 4.02 S|********------>      .      .      .|
|      10:15 10:30  DEP PRIV PAGE  2.33 S|****----->    .      .      .      .|
|      10:30 10:45  DB PMFG021     5.25 S|********----------> .      .      .|
|      10:45 11:00  DB PMFG021     8.90 S|**************************----->   .|
|      11:00 11:15  DEP PRIV PAGE  2.89 S|****------->  .      .      .      .|
|      11:15 11:30  DB PMFG021     5.38 S|*************------> .      .      .|
|      11:30 11:45  DB PACCT03     2.25 S|***----->     .      .      .      .|
|      11:45 12:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|08/31          .                        |                                    |
|               .                        |                                    |
|               .                        |                                    |
+=============================================================================+
```

Figure 34. Single-interval summary display

In its default single-interval form, this report is too long to be of general use. Instead of using the long form, you can specify that the data be combined into larger intervals than the 15-minute RMF interval. This interval allows reporting a long time on a single report.

Multi-interval summary display

The following example shows how the COMBINE command combines four 15-minute single intervals to produce one line on the report. Each line represents 1 hour instead of 15 minutes.

The command:

```
DISPLAY SUMMARY SYS SDATE(08/29/15) STIME(10) -
EDATE(09/02/15) ETIME(12) COMBINE(1H)
```

generates the report that is shown in Figure 35 on page 50.

```
+==============================================================================+
| SYSTEMWIDE                                                                   |
| Period: 10:00 to 12:00 on 08/29/15 to 09/02/15                               |
+------------------------------------------------------------------------------+
|                           COMPETING TRANSACTIONS                             |
+------------------------------------------------------------------------------+
|DATE__START_END____MAIN_REASON___TIME___|0S_____2______4______6______8_____10|
|08/29 10:00 11:00  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      11:00 12:00  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
|08/30 10:00 11:00  DB PMFG021     7.06 S|*****************-------->  .      .|
|      11:00 12:00  DB PMFG021     9.85 S|*********************-------------->.|
|08/31 10:00 11:00  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
|      11:00 12:00  DB PMFG021     5.38 S|*************------> .      .      .|
|09/01 10:00 11:00  DB PACCT03     1.73 S|***--->.      .      .      .      .|
|      11:00 12:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|09/02 10:00 11:00  DEP PRIV PAGE  3.40 S|*****************------>    .      .|
|      11:00 12:00  DB PMFG021     8.80 S|********************----------->   .|
+==============================================================================+
```

Figure 35. Multi-interval summary display

The START and END columns reflect the start and end times of the RMF cycle combined into 1-hour intervals.

RCPU CPU activity resource panel

If you do not provide a multi-interval length with the COMBINE keyword, then all the intervals are combined into one display.

For example, to see a COMBINEd resource display of CPU activity from 9:00 AM to 5:00 PM, use the RCPU keyword and enter:

```
DISPLAY  RCPU  YESTERDAY STIME(9) ETIME(17) COMBINE
```

to generate the report that is shown in Figure 36 on page 50.

```
+=============================== CPU Activity =================================+
| Period:  09:00 to 17:00 on 08/29/15                  Elap = 08:00 H    IMSD |
| COMBINATION OF 8 INTERVALS                                                   |
+------------------------------------------------------------------------------+
| Region   Type   TCB    SRB   Total    | Region   Type   TCB    SRB   Total   |
| ------   ----   ---    ---   -----    | ------   ----   ---    ---   -----   |
| CONTROL  CTL   17.2 %  4.4 %  21.6 % | MESSAGE4 MPP    5.2 %  1.2 %   6.4 % |
| MESSAGE1 MPP    3.8 %   .9 %   4.7 % | FPMSG1   FP     9.7 %  3.2 %   2.9 % |
| MESSAGE2 MPP   11.7 %  2.0 %  13.7 % | FPMSG3   FP     1.0 %   .3 %   1.3 % |
| MESSAGE3 MPP    4.5 %  1.1 %   5.6 % |                                      |
+==============================================================================+
```

Figure 36. RCPU CPU Activity resource panel

In the CPU activity panel that is shown here, the total for each region in relation to the total CPU time is given. If the totals are added, you see the percentage of total CPU time used by the IMS environment.

The region totals for the indicated period (8 hours in this case) are further broken down to TCB and SRB percentages.

Multi-interval detail display (AVERAGE)

The length of the time interval that is specified as a subparameter of COMBINE is generally an even multiple of your RMF collection interval length. Using an interval that is not an even multiple, while allowed, can cause displays that are irregular, confusing, or difficult to use. Also, do not use an interval that is less than your RMF collection interval.

The miscellaneous option PLOTMIN(3) applies only to DETAIL degradation displays; it means that only those wait reasons that account for more than three percent of the total transaction displays. It is only displayed because PLOTMIN is always set as a default; it is ignored by all resource displays.

Indiscriminate use of COMBINE can cause EPILOG to read and process a large amount of data. When you combine weeks' or months' worth of data together, you might want to use EPILOG in batch mode.

When you use DETAIL and SUMMARY displays over the COMBINEd multi-interval, the default display is the AVERAGE response or elapsed times over the entire multi-interval. For example:

```
DISPLAY  GRP 3 SDATE(08/29/15) STIME(0900)
EDATE(08/29/15) ETIME(12) COMBINE PMIN(3)
```

generates the report that is shown in Figure 37 on page 51.

```
-----------------------------------------------------------------------------
|                                                                             |
| Workloads:    Transaction Group 3                                           |
| Periods:      Starting 08/29/08 at 09:00 and ending 08/29/15 at 12:00      |
| Misc Options: COMBINE AVERAGE PLOTMIN(3)                                    |
|                                                                             |
 -----------------------------------------------------------------------------
| Workload:     TRANSACTION GROUP 3                                           |
| Period:       09:00 to 12:00 on 08/29/15              Elap =   3 H   IMSD   |
| Average of 12 intervals                                                     |
+=============================================================================+
|                            RESPONSE TIME DATA                               |
+=============================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue        |       0.61s       |      174      |       13.38         |
|Pgm Input Queue    |       0.00s       |        0      |         .00         |
|Processing         |       2.05s       |      173      |       13.31         |
|Response Time 0    |       2.66s       |      173      |       13.31         |
|Output Queue       |       0.00s       |       54      |        4.15         |
|Response Time 1    |       3.53s       |       54      |        4.15         |
+=============================================================================+
|                               BOTTLENECK DATA                               |
+=============================================================================+
|COMPETING_STATE_________TIME_____%_|0___1___2___3___4___5___6___7___8___9___0|
|Scheduling Waits      6.35 S   63.0|--------------===========>  .   .   .   .|
| Wait for Resched   (2.75) S (29.6)|------------>.   .   .   .   .   .   .   .|
| Wait for MPP       (2.30) S (20.4)|---------->  .   .   .   .   .   .   .   .|
| Wait for GU        (1.30) S (13.0)|----->  .   .   .   .   .   .   .   .   .|
|z/OS Waits            3.46 S   31.5|------------=>  .   .   .   .   .   .   .|
| PGM Fetch I/O      (1.31) S  (9.9)|--->.   .   .   .   .   .   .   .   .   .|
| BLDL I/O           (2.15) S (21.6)|---------->  .   .   .   .   .   .   .   .|
|IMS Waits             1.15 S    5.5|--> .   .   .   .   .   .   .   .   .   .|
| IRLM Conflict Wait (1.15) S  (5.5)|--> .   .   .   .   .   .   .   .   .   .|
|-----------------------------------|                                         |
|Average (res time 0)  2.66 S       |                                         |
+=============================================================================+
```

Figure 37. Multi-interval detail display (AVERAGE)

Combined interval detail display (TOTAL)

The example in "Multi-interval detail display (AVERAGE)" on page 51 averages 12 intervals, which covers the time period between 9:00 and noon. The average of response time 0 is 2.66 seconds. You can make the same request but include the TOTAL keyword:

```
DISPLAY GRP 3 SDATE(08/29/15) STIME(0900)
EDATE(08/29/15) ETIME(12) COMBINE TOTAL
```

to generate the report shown in Figure 38 on page 52.

```
-----------------------------------------------------------------------------
|                                                                             |
| Workloads:    Transaction Group 3                                           |
| Periods:      Starting 08/29/15 at 09:00 and ending 08/29/15 at 12:00       |
| Misc Options: COMBINE TOTAL PLOTMIN(5)                                      |
|                                                                             |
 -----------------------------------------------------------------------------
| Workload:     TRANSACTION GROUP 3                                           |
| Period:       09:00 to 12:00 on 08/29/15              Elap =   3 H   IMSD   |
| Average of 12 intervals                                                     |
+-----------------------------------------------------------------------------+
|                          RESPONSE TIME DATA                                 |
+-----------------------------------------------------------------------------+
|RESPONSE_COMPONENT____TOTAL RSP. TIME______TRANS. COUNT_____RATE (PER. MIN.) |
|INPUT QUEUE         |     3:20 H        |     41746       |       36.92      |
|PGM INPUT QUEUE     |    19:32 M        |      6769       |        5.99      |
|PROCESSING          |     6:46 H        |     48513       |       42.90      |
|RESPONSE TIME 0     |    10:26 H        |     48513       |       42.90      |
|OUTPUT QUEUE        |     3:21 H        |     42237       |       37.35      |
|RESPONSE TIME 1     |    10:08 H        |     41608       |       36.80      |
+-----------------------------------------------------------------------------+
|                             BOTTLENECK DATA                                 |
+-----------------------------------------------------------------------------+
|COMPETING_STATE_________TIME_____%_|0___1___2___3___4___5___6___7___8___9___0|
|SCHEDULING WAITS       2:55 H   71.6|------------===========>>>>>   .   .   .|
| WAIT FOR RESCHED   (8:16) M  (6.9)|--> .   .   .   .   .   .   .   .   .   .|
| WAIT FOR MPP       (8.34) M  (6.9)|--> .   .   .   .   .   .   .   .   .   .|
| WAIT FOR GU        (2:30) H (61.3)|------------===========>   .   .   .   .|
|DATABASE I/O WAITS     3:04 H   12.9|----->  .   .   .   .   .   .   .   .   .|
|z/OS WAITS             1:14 H    5.2|--> .   .   .   .   .   .   .   .   .   .|
|IMS WAITS              1:49 H    7.6|--->.   .   .   .   .   .   .   .   .   .|
|-----------------------------------                                          |
|TOTAL (RESP TIME 0)  10:26 H                                                 |
+=============================================================================+
```

Figure 38. Combined interval detail display (TOTAL)

The example is similar to the average display, however, the TOTAL display gives total, rather than average, transaction times.

You can combine any two or more of the available resource panels.

Multiple performance group bottleneck report

At times, you might combine data from several transaction groups and treat them as a single workload. For example, your installation might have several transaction groups that are defined to separate various tasks, but a requirement might still exist for a report that crosses over the group boundaries.

You can instruct the EPILOG reporter to combine several groups and treat them as one. For example, suppose you have three defined as 2, 4, and 12. You can combine these three groups by specifying them all as operands to the GRP keyword:

```
DISPLAY  GRP (2,4,12) TODAY STIME(14) ETIME(1415) PMIN(3)
```

to generate the report that is shown in Figure 39 on page 53.

```
 ----------------------------------------------------------------------------
|                                                                            |
| Workloads:    Transaction Group(2 4 12)                                    |
| Periods:      Starting on 8/30/15 at 14:00 and Ending on 8/30/15 at 14:15  |
| Misc Options: SINGLE PLOTMIN(3)                                            |
|                                                                            |
 ----------------------------------------------------------------------------
+============================================================================+
|                            RESPONSE TIME DATA                              |
+============================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue        |      0.74s       |       79       |       5.27         |
|Pgm Input Queue    |      0.33s       |       78       |       5.20         |
|Processing         |      8.78s       |       78       |       5.20         |
|Response Time 0    |      9.85s       |       78       |       5.20         |
|Output Queue       |      0.10s       |       76       |       5.07         |
|Response Time 1    |      9.99s       |       76       |       5.07         |
+============================================================================+
|                               BOTTLENECK DATA                              |
+============================================================================+
|COMPETING_STATE_________TIME_____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU             1.77 S   18.0|------->.   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.78) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS   (0.99) S (10.1)|---->   .   .   .   .   .   .   .   .   .|
|Scheduling Waits       .54 S   10.9|---->   .   .   .   .   .   .   .   .   .|
| Wait for MPP       (0.54) S (10.9)|---->   .   .   .   .   .   .   .   .   .|
|Database I/O Waits    6.12 S   62.1|------------=============>  .   .   .   .|
| PMFG021            (5.82) S (59.0)|------------===========>.   .   .   .   .|
| BE3INDX            (0.30) S  (3.1)|->  .   .   .   .   .   .   .   .   .   .|
|z/OS Waits            0.44 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP) (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|IMS Activity          0.44 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL   (0.44) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|AVERAGE (Rtime 0)   9.85 S         |                                     .   .|
+============================================================================+
```

Figure 39. Multiple performance group bottleneck report

This feature can be used with either DETAIL or SUMMARY reports.

Chapter 5. Selecting EPILOG displays based on bottleneck factors

The following topics present the exception keywords that you can use with the DISPLAY command and the exceptions that you can use with combined intervals.

Exception filtering

Sometimes, because of specific bottleneck factors, you want to focus only on specific time intervals. EPILOG exception analysis techniques are an efficient way to display only intervals, which meet certain selection criteria.

It can be tedious to search through hundreds of collection intervals and look for periods when response time is poor or when a particular type of bottleneck occurs. To relieve this problem, the DISPLAY command accepts exception filters that display intervals that meet certain selection criteria.

For example, you might want to see only intervals where response time was greater than 2 seconds, or where database I/O waits accounted for more than 20% of the total lifetime of a transaction.

Exception filtering is optional. If you do not provide an exception clause as part of the DISPLAY command, all intervals within the specified time limits are processed and display.

If you specify multiple exceptions, all limits must be met for an interval to be selected (the requests are achieved by a logical AND calculation).

Syntax for bottleneck factors

The following tables define the bottleneck factors that you can use for the following components:

- CPU usage
- Scheduling waits
- Database I/O waits
- z/OS waits
- IMS waits
- External Subsystem (ESS) waits
- Output waits
- Response time exceptions

Table 13. CPU usage

Keyword	Short	Description
CPU usage		
CPUAPPL	APL	Using CPU in application
CPUIMS	IMS	Using CPU in IMS
CPUTIL	CPA	Total CPU usage

Table 14. Scheduling waits

Keyword	Short	Description
Scheduling waits		
ALOPSBLT	APS	TM Allocate PSB latch

Table 14. Scheduling waits (continued)

Keyword	Short	Description
BLKLOAD	BLK	Block loader
BLOCKMVR	BMW	Block mover wait
DDIRBLAT	DDB	DDIR Block latch
DDIRPLAT	DDP	DDIR Pool latch
DMBBLTCH	DBL	DMB Block latch
DMBPOOL	DMB	DMB pool
GETUNIQ	WGU	Wait for get unique
INTENTCN	INC	Intent conflict
PDIRBLAT	RBL	PDIR Block latch
PDIRPLAT	PPL	PDIR Pool latch
PSBBLTCH	PBL	PSB Block latch
PSBPOOL	PSB	PSB pool
PSBWPOOL	PSW	PSBW pool
SCHEDBLK	SCB	Scheduling blocked
SCHEDCMP	SCC	Total scheduling wait for competing transactions
SCHEDEXC	SCE	Total scheduling wait for executing transactions
SCHEDULE	SCA	All scheduling waits
TMSCHED	TSL	TM Scheduling latch
UNSCHED	UNS	Cannot be scheduled
WAITBMP	WBM	Wait for BMP
WAITIFP	WIF	Wait for IFP
WAITMPP	WMP	Wait for MPP
WAITRES	WRE	Wait for reschedule

Table 15. Database I/O waits

Keyword	Short	Description
Database I/O waits		
DATABASE	DBW	Database I/O waits

Table 16. z/OS waits

Keyword	Short	Description
z/OS waits		
APPLIO	API	Application I/O
BLDLIO	BLD	BLDL I/O
CPUWTCTL	CTL	Control region CPU wait

Table 16. z/OS waits (continued)

Keyword	Short	Description
CPUWTDEP	CPW	CPU wait (MPP/BMP)
CSAGFA	GCS	CSA page (GFA)
CSAMSDB	CMS	CSA page (MSDB)
CSAPAGE	CSA	CSA page
CTLEPAGE	CEP	CTL extended private page
CTLPAGE	CPP	CTL private page
CTLPGFA	GPP	CTL private page (GFA)
DEPEPAGE	DEP	Dependent extended private page
DEPPAGE	DPP	Dependent private page
DEPPGFA	GDP	Dependent private page (GFA)
ECSAMSD	EMS	ECSA page (MSDB)
EXTCSA	ECS	ECSA page
EXTLPA	ELP	Extended LPA page
LPAGFA	GLP	LPA page (GFA)
LPAPAGE	LPA	LPA page
z/OSWAITS	WMV	z/OS waits
PGMFETCH	PGF	Program fetch I/O
SWAPPIN	SWP	Swapping in
XMGFA	GXM	Cross memory page (GFA)
XMPAGE	XMP	Cross memory page

Table 17. IMS waits

Keyword	Short	Description
IMS waits		
ACTLLTCH	ACL	ACL
ADSCDLT	ADL	ADL
AUTHLTCH	AUL	AUL
ACTLLTCH	ACL	ACL
BLKLREL	BLR	Block Loader/Relocate Wait
CBTLATCH	CBT	CBT latch
CONVCKPT	CCP	Conversation Checkpoint latch
DBBLATCH	DBB	DBB latch
DBLKLTCH	DLK	DBLK latch
DBSYCKPT	DBS	Database (DB) System Checkpoint latch
DCSYCKPT	DCS	Data communications (DC) System Checkpoint latch

Table 17. IMS waits (continued)

Keyword	Short	Description
DEDBARL	DAL	DEDB area lock
DEDBSEG	DES	DEDB segment
DLIWAIT	DLI	Other DL/I IWAIT
DLOGLTCH	DLL	DLOG latch
DMACLAT	DML	DMAC latch
DMACSLT	DSL	DMAC share latch
DMELATCH	DME	DME latch
FNCBLAT	FNL	FNCB latch
FPABEND	FOA	Fast Path other abend
FPBUFFR	FBF	Fast Path buffer
FPITERM	FIT	Fast Path IWAIT in term
FPPSABN	PAB	Fast Path pseudo abend
FPSYNCP	FSY	Fast Path SYNC processing
FPSYNCL	FSL	Fast Path SYNC lock
GCMDLTCH	GCL	GCMD latch
GENRLTCH	GNL	GENR latch
HDSLATCH	HDS	HDS latch
HSWAIT	HSP	Wait for HSSP pool space
IMSDISP	IWD	IWAIT in IMS dispatcher
IMSWAIT	WIM	Total IMS waits
INQLATCH	INQ	INQ latch
IRWAIT	IRW	IRLM conflict wait
ISWITFP	IFT	ISWITCHED to Fast Path TCB
ISWTLSO	ISC	ISWITCHED to CTL
LGMSGIO	ISL	ISWITCHED to LSO
LOGLLTCH	LMS	Long MSG I/O
LQBLATCH	LGL	LOGL latch
LUMLATCH	VLQ	LQB Pool latch
MFSLOAD	LUM	LU 6.2 Manager (LUM) latch
MSDBLAT	MFS	MFS load
MSDBSEG	MSL	MSDB latch
NDQLATCH	MSS	MSDB segment
OBAALLO	NDQ	NDQ latch
OBALOCK	OBA	Fast Path overflow buffer allocation

Table 17. IMS waits (continued)

Keyword	Short	Description
OPCLLAT	FOB OCL	Fast Path overflow buffer lock Open/close latch
OBFLATCH	OBF	Wait for an OBFM latch
OSMBUFF	OSB	Wait for an OSAM buffer
OTABEND	ABN	Other ABENDs
OTHRLTCH	OTL	OTHR latch
PIWAIT	PIW	Program isolation wait
PSABEND	PSA	Psuedo abend
QBFLATCH	QBF	QBF latch
QBLKSIO	QBL	QBLKS I/O
QMGRLTCH	QMR	Queue Manager latch
RESRLAT	FRL	Fast Path resource latch
SHMSGIO	SHM	SHMSG I/0
SMBQLTCH	SMB	SMB Queue latch
SMGTLTCH	SML	SMGT latch
SNDLATCH	SND	SND latch
SPAIO	SPA	SPA I/O
SUBQLTCH	SBQ	TM Subqueue latch
SYNCWAIT	SPW	Sync point wait
TCTLATCH	TBL	TCT Block latch
TERMINAL	TML	DC Terminal latch
TERMWAIT	IWT	IWAIT in termination
USERLTCH	USR	DC User latch
VBFLATCH	VBF	VBF latch
VSMBUFF	VSB	Wait for a VSAM buffer
VTCBLTCH	VTC	VTCB Pool latch
XCNQLTCH	XCL	XCNQ latch

Table 18. External Sub-system (ESS) waits

Keyword	Short	Description
External Sub-system (ESS) waits		
ESSCRT	CRT	Create thread
ESSSQL	SQL	SQL call
ESSTRM	TRM	Terminate thread
ESSWAIT	ESS	Total ESS waits

Table 19. Output waits		
Keyword	**Short**	**Description**
Output waits		
FPOUTMS	MOQ	Fast path output waits
OTHERW	OTW	Other waits
OUTPUT	OUT	Output waits

Table 20. Response time exceptions		
Keyword	**Short**	**Description**
Response time exceptions		
INPUTQC	IQC	Number of transactions with input queue time
INPUTQR	IQR	Transaction rate for input queue
INPUTQT	IQT	Input queue time
OUTPUTQC	OQC	Number of transactions with output queue time
OUTPUTQR	OQR	Transaction rate for output queue
OUTPUTQT	OQT	Output queue time
PGRMQC	PQC	Number of transactions with program input queue time
PGRMQR	PQR	Transaction rate for program input queue
PGRMQT	PQT	Program input queue time
PROCESSC	POC	Number of transactions processed
PROCESSR	POR	Transaction rate for processing
PROCESST	POT	Processing time
R0COUNT	R0C	Number of transactions with response time 0
R0RATE	R0R	Transaction rate for response time 0
R0TIME	R0T	Response time 0
R1COUNT	R1C	Number of transactions with response time 1
R1RATE	R1R	Transaction rate for response time 1
R1TIME	R1T	Response time 1

The standard exception format for either the REPORTIF or SELECTIF keyword follows:

```
REPORTIF( keyword1 (options (pnnq)) keyword2 (options (pnnq)) ... )
```

keyword
: refers to the performance measure being filtered, such as response time or database I/O waits (DBW). See Table 16 on page 56 for a list of the available options.

options
: used only with I/O and enqueue exceptions. See Table 9 on page 40 for a list of available options.

pnnq
: selection criteria for the exception

p
: > greater than (default) < for less than

nn
: decimal number

q
: type (default is %)

%
: percent

S
: seconds

M
: minutes

H
: hours

Examples:

<1.5S
: less than 1.5 seconds

<10
: less than 10 percent

<10%
: less than 10 percent

10M
: greater than 10 minutes

>56S
: greater than 56 seconds

10
: greater than 10 percent

Here is an example of using the exception format. To report on all intervals where the scheduling-wait time for transaction group 2 was greater than 2 seconds, you would use the following command.

```
DISPLAY GRP(2) TODAY  REPORTIF( SCHEDULE(>2S) )
```

Note that you can specify more than one filter in a single command. The following example displays all those intervals where output wait time was greater than .5 seconds and database I/O waits made up greater than 30 percent of the total transaction waits:

```
DISPLAY GRP(2) TODAY  RIF( OUTPUT(>.5S) DBW(>30%) )
```

If you wanted to display all intervals for which *no* waits of a certain type recorded you would prefix the exception type keyword with NO and give no numeric arguments (pnnq). For example, **NOWMP** is the equivalent of **WMP(<0%)**.

Exceptions with combined intervals (SELECTIF keyword)

The exception filtering that is started by the REPORTIF or RIF exception keyword completes just before the interval is printed.

When you use multi-interval displays, where multiple collection records are averaged together, the intervals are combined and then RIF processing is done. For example, if your installation has 15-minute RMF intervals, you might decide to combine them into 1-hour multi-intervals and display those intervals in which scheduling wait time exceeds 2 seconds:

```
DISPLAY GRP(2) TODAY COMBINE(1H)  RIF( SCHEDULE(>2S) )
```

RIF processing treats multi-intervals in the same way as it does single intervals.

There is another type of exception processing available; you start it by using the SELECTIF or SIF keyword. SIF filtering is done at the RMF interval level rather than the multi-interval level. Consider the same example but this time, SIF is used rather than RIF exception processing:

```
DISPLAY GRP(2) TODAY COMBINE(1H)  SIF( SCHEDULE(>2S) )
```

In this example, EPILOG selects only for COMBINEing those 15-minute RMF intervals where the scheduling waits is greater than 2 seconds; the 1 hour multi-interval might be made up of anywhere from zero to four real intervals. If no RMF intervals pass the test, nothing displays for the 1-hour multi-interval.

REPORTIF and SELECTIF can be viewed in this way: REPORTIF reports data after it has EPILOG combines all data and then the REPORTIF keyword filters them. SELECTIF determines whether the data is selected for inclusion in a multi-interval.

SIF and RIF *can* be used together in the same request. We recommend that do not use SIF and RIF in the same request until you become proficient in coding exception requests because dual filtering can get confusing.

```
DISPLAY GRP(2) TODAY COMBINE(1H)  SIF( SCC(>2S) ) RIF( DBW(>20%) )
```

This example first selects any interval that has scheduling wait times greater than 2 seconds and combines these intervals into 1-hour multi-intervals. These multi-intervals would be displayed only if database I/O waits made up more than 20 percent of the total transaction times.

SIF and RIF give the same results unless you are combining RMF intervals; if you request single-interval displays, it does not matter which keyword you use.

The following flowchart might be helpful in understanding the differences between SIF and RIF.

```
            -------------------
            | Read another    |
----------->| record from the |<---------- <----<-----
|           | EDS             |            |    |    |
|           ---------^---------            |    |    |
|                    |                     |    |    |
|                    !                     |    |    |
|           -------------------            |    |    |
|           | Apply SELECTIF  | Ignore     |    |    |
|           | filter (if any) {-------------    |    |
|           | to the record   |                 |    |
|           ---------^---------                 |    |
|                    |                          |    |
|                    ! Pass                     |    |
|           ------------------                  |    |
|           | COMBINE record |                  |    |
|           | into a multi-  |                  |    |
|           | interval       |                  |    |
|           ---------^--------                  |    |
|                    |                          |    |
|                    !                          |    |
|           -------------------                 |    |
|       Yes | Are there more  |                 |    |
------------} records to be   |                 |    |
            | COMBINEd?       |                 |    |
            ---------^---------                 |    |
                     |                          |    |
                     ! No                       |    |
            -------------------                 |    |
            | Apply REPORTIF  | Ignore          |    |
            | filter (if any) {------------------    |
            | to the multi-   |                      |
            | interval        |                      |
            ---------^---------                      |
                     |                               |
                     ! Pass                          |
            -------------------                      |
            | DISPLAY the data|                      |
            | (if any have    {-----------------------
            | survived)       |
            -------------------
```

Figure 40. REPORTIF and SELECTIF exception processing

COMBINEd records are kept in memory and are not part of the EDS. The records are built and displayed as combined intervals so that the loop back to the EDS from the REPORTIF block is issued after the multi-interval is tested. Resources and processing time are used more efficiently.

Chapter 6. EPILOG system navigator

The following topics summarize the basic types of EPILOG displays and describe the system navigator. The system navigator is an EPILOG facility that helps you investigate periods of poor system response time and bottlenecks.

EPILOG displays

EPILOG provides several basic types of displays.

Collection Options (O)
You can used this option to determine quickly the workload whose response time and bottleneck information is being displayed.

GROUPSUMMARY (GSUM)
The GROUPSUMMARY display shows the transaction rate and average response time by transaction group.

SUMMARY
The SUMMARY display produces one line of output for each interval or multi-interval. Each line shows the most significant bottleneck along with a plot of the average transaction response time.

DETAIL
The DETAIL bottleneck display provides an in-depth breakdown of bottlenecks for an interval or multi-interval.

Resources (Rccc commands)
These are pre-defined displays, which contain resource information that is extracted from RMF and EPILOG data sampling, such as CPU activity, DASD activity, and scheduling waits.

Each of these displays can use the collected data at two levels of summarization:

SINGLE
This means that the data is shown either by the RMF interval or by your defined collection interval.

COMBINE
COMBINEd displays are those displays that take data for more than one RMF interval and average them in some meaningful fashion.

The EPILOG displays show the following basic types of panels:

- **Multi-interval GSUM** - Example:

  ```
  DISPLAY GSUM TODAY COMBINE STIME(1100) ETIME(1115)
  ```

- **Multi-interval summary** - Example:

  ```
  DISPLAY GRP(2) TODAY SUMMARY COMBINE(1H)
  ```

- **Multi-interval detail report** - Example:

  ```
  DISPLAY GRP(2) TODAY COMBINE STIME(1100) ETIME(1200)
  ```

- **Multi-interval resource displays** - Example:

  ```
  DISPLAY RCPU TODAY COMBINE STIME(1100) ETIME(1200)
  ```

- **Single-interval GSUM** - Example:

  ```
  DISPLAY GSUM TODAY STIME(1100) ETIME(1200)
  ```

- **Single-interval summary** - Example:

  ```
  DISPLAY GRP(2) TODAY SUMMARY STIME(1100) ETIME(1200)
  ```

- **Single-interval detail report** - Example:

```
DISPLAY GRP(2) TODAY STIME(1115) ETIME(1130)
```

- **Single-interval resource displays** - Example:

```
DISPLAY RCPU TODAY STIME(1115) ETIME(1130)
```

You can use the EPILOG system navigator to easily explore this tree of possible displays in fullscreen mode and implement IBM's logical tuning technique. (See "Approach to performance tuning with EPILOG" on page 4 if you are not familiar with this procedure.) Here is an example of the way to use the system navigator.

In this example, assume that the percentage threshold is set to 3 with the command:

```
SET PMIN(3)
```

As shown in the next figure, entering an *S* next to transaction group 7 produces a summary display of the same data:

```
+============================================================================+
 | Group Summary                                                             |
 | Period:  10:29 to 10:44 on 08/29/15                    Elap = 15:00 M IMST  |
 +---------------------------------------------------------------------------+
 |                          COMPETING TRANSACTIONS                           |
 +---------------------------------------------------------------------------+
 |                RESPONSE TIME 0 VALUES |                                    |
 |Group___Name______Rate___Count____Time__|0S_____2______4______6______8_____10|
 |      System      4.67      70    0.99 S|--->   .      .      .      .      .|
 |      Other        .00       0    0.00 S|.      .      .      .      .      .|
 |  01  Billing     0.07       1    1.49 S|-----> .      .      .      .      .|
 |  02  Shipping    1.93      29    1.21 S|---->  .      .      .      .      .|
 |  04  Account    11.28      53    1.00 S|-->    .      .      .      .      .|
 |  05  Payroll     2.13      30    0.41 S|>      .      .      .      .      .|
 |  06  Inventory   3.04      34    1.15 S|--->   .      .      .      .      .|
S  07  Orderproc   2.80      42    6.91 S|---------------------->    .      .|
 |----------------------------------------                                    |
 |Average (Resp Time 0)            1.01 S                                    |
 +===========================================================================+


 EPILOG            08/29/15     11:49  Mode: PAGE     D1              LAST FRAME
 CMD==>
 +===========================================================================+
 | Transaction Group =  7                                                    |
 | Period:  10:29 to 10:44 on 08/29/15                                       |
 +---------------------------------------------------------------------------+
 |                          COMPETING TRANSACTIONS                           |
 +---------------------------------------------------------------------------+
 |Date__Start_End___Main_Reason(*)_Time(-)|0S_____2______4______6______8_____10|
 |08/29 10:29 10:44 Wait for MPP    6.91 S|*****------------------>    .       |
 |----------------------------------------                                    |
 |Average (Resp Time 0)            6.91 S                                    |
 +===========================================================================+
```

Figure 41. GSUM to SUM

Entering *D* on the vertical bar next to the specified group displays detail data. Entering *R* next to the specified group displays resource data.

To generate a multi-interval summary report for a day from 9:00 AM to 9:00 PM, enter the following command:

```
DISPLAY GRP 2 SUM SDATE(08/29/15) STIME(09:00) -
EDATE(08/29/15) ETIME(21:00) COMBINE(1H)
```

the report displays as shown in Figure 42 on page 67.

```
 ---------------------------------------------------------------------------
|                                                                           |
| Workloads:    TRANSACTION GROUP 2                                         |
| Periods:      Starting on 08/29/15                                        |
| Misc Options: SUMMARY COMBINE(60M) AVERAGE                                |
|                                                                           |
 ---------------------------------------------------------------------------
+===========================================================================+
| Transaction group 2                                                       |
| Period: 09:00 to 21:00 on 08/29/15                                        |
+---------------------------------------------------------------------------+
|DATE__START_END____MAIN_REASON___TIME___|0S_____2______4______6______8_____10|
|08/29 09:00 10:00  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      10:00 11:00  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
|      11:00 12:00  DB PMFG021     7.06 S|*****************-------->  .      .|
S      12:00 13:00  DB PMFG021     9.85 S|*********************------------->.|
|      13:00 14:00  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
|      14:00 15:00  DB PMFG021     5.38 S|*************------> .      .      .|
|      16:00 17:00  DB PACCT03     1.73 S|***--->.      .      .      .      .|
|      17:00 18:00  DB PMFG021     2.94 S|***------->   .      .      .      .|
|      19:00 20:00  DEP PRIV PAGE  6.40 S|*****************------>    .      .|
|      20:00 21:00  DB PMFG021     8.80 S|********************----------->   .|
+===========================================================================+
```

Figure 42. Multi-interval summary display

The format of the SUMMARY report makes it obvious that response time got appreciably worse between noon and 13:00. The next step might be to ask for the single-interval breakdown of the SUMMARY report, by entering:

```
DIS GRP(2) SUMMARY SDATE(08/29/15) STIME(12) EDATE(08/29/15) ETIME(13)
```

You can use the system navigator to enter an *S* (for select single-interval display) in the first position of the line. (Note the **S** in Figure 42 on page 67.) A command is internally generated to produce a single-interval display:

```
+---------------------------------------------------------------------------+
|                                                                           |
| Workloads:    Transaction Group 2                                         |
| Periods:      Starting on 08/29/15                                        |
| Misc options: SUMMARY SINGLE                                              |
|                                                                           |
|___________________________________________________________________________|
| WORKLOADS:    TRANSACTION GROUP 2                                         |
| PERIOD: 08/29/15 12:00     TO   08/29/15 13:00                            |
+===========================================================================+
|DATE__START_END____MAIN_REASON___TIME___|0S_____2______4______6______8_____10|
|08/29 12:00 12:15  USING CPU/APPL 2.02 S|***---->      .      .      .      .|
|      12:15 12:30  DEP PRIV PAGE  1.84 S|****--->      .      .      .      .|
D      12:30 12:45  DB PMFG021     9.85 S|*********************------------->.|
|      12:45 13:00  DEP PRIV PAGE  2.26 S|****---->     .      .      .      .|
+===========================================================================+
```

Figure 43. Summary report of IMS response time

A brief examination shows us that the major problem was in the 12:30 to 12:45 PM time interval. At this point, you might want to see a detailed display of bottlenecks during this interval. You can enter the command:

```
DIS GRP(2) SDATE(08/29/15) STIME(1230) EDATE(08/29/15) ETIME(1245)
```

You can use the system navigator simplifies to enter a *D* (for detailed bottleneck report) in the first column of the appropriate line. (Note the **D** in Figure 43 on page 67.) The detailed display is shows as follows:

```
+==============================================================================+
|                                                                              |
| Workloads:    Transaction Group 2                                            |
| Periods:      Starting 12:30 on 08/29/15 and ending 12:45 on 08/29/15        |
| Misc options: DETAIL PLOTMIN(3)                                              |
|                                                                              |
+------------------------------------------------------------------------------+
| WORKLOAD:     TRANSACTION GROUP 2                                            |
| PERIOD: 12:30 to 12:45 on 08/29/15                  Elap =   15:00 M  IMSD   |
+==============================================================================+
|                            RESPONSE TIME DATA                                |
+==============================================================================+
|RESPONSE_COMPONENT____AVG. RSP. TIME____TRANS. COUNT____RATE (per. min.)     |
|Input Queue         |      0.74s       |       79      |       5.27           |
|Pgm Input Queue     |      0.33s       |       78      |       5.20           |
|Processing          |      8.78s       |       78      |       5.20           |
|Response Time 0     |      9.85s       |       78      |       5.20           |
|Output Queue        |      0.10s       |       76      |       5.07           |
|Response Time 1     |      9.99s       |       76      |       5.07           |
+==============================================================================+
|                               DEGRADATION DATA                               |
+==============================================================================+
|COMPETING_STATE________TIME____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU             1.77 S   18.0|------->.   .   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.78) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS   (0.99) S (10.1)|---->   .   .   .   .   .   .   .   .   .   .|
|Scheduling Waits      0.54 S   10.9|---->   .   .   .   .   .   .   .   .   .   .|
| Wait for MPP       (0.54) S (10.9)|---->   .   .   .   .   .   .   .   .   .   .|
RDatabase I/O Waits    6.12 S   62.1|------------============>  .   .   .   .   .|
| PMFG021            (5.82) S (59.0)|------------===========>.  .   .   .   .   .|
| BE3INDX            (0.30) S  (3.1)|-> .   .   .   .   .   .   .   .   .   .   .|
|z/OS Waits            0.44 S    4.5|-> .   .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP) (0.44) S  (4.5)|-> .   .   .   .   .   .   .   .   .   .   .|
|IMS Activity          0.44 S    4.5|-> .   .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL   (0.44) S  (4.5)|-> .   .   .   .   .   .   .   .   .   .   .|
|----------------------------------|                                       .   .|
|TOTAL (Resp Time 0)   9.85 S       |                                       .   .|
+==============================================================================+
```

Figure 44. DETAIL report example

You might want to see some resource data for the database I/O waits bottleneck. PMFG021 was the database with the poorest response time. Resource data is displayed by the resource panel or panels that are associated with the bottleneck reason. For example, to see the resources that are associated with databases, you can issue the following request:

```
DIS RDEV SDATE(08/29/15) STIME(12:30) EDATE(08/29/15) ETIME(12:45)
```

To request resource data with the system navigator, you enter an *R* in the first position of the line that shows the bottleneck. (Note the **R** in Figure 44 on page 68.) The result is a display that is similar to the following output:

```
+============================== IMS Device Statistics ==========================+
| Period:  12:30 to 12:45 on 08/29/15                       Elap =15:00 M   IMSD |
+--------------------------------------------------------------------------------+
|volser|Unit|  SSCHs |  SIOs   | %Dev| Avg q | Resp= IOSQ+ Pend+ Conn+ Disc| Open|
|      |    |per sec | per sec| util| Length| Time| Time| Time| Time| Time| Dsns|
+------+----+--------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|PRD817| 102|      .9|      .0| 23.5|   .05 | 23.5|   .1|  1.0| 10.6| 11.8| 10.8|
|TSO802| 142|     2.2|      .0| 18.5|   .05 | 18.5|   .0|   .2|  3.4| 14.9| 14.3|
|TSO812| 14C|      .7|      .0| 23.8|   .01 | 23.8|   .0|   .2| 10.1| 13.5| 12.3|
|IPS801| 150|      .4|      .0| 15.7|   .00 | 15.7|   .0|   .0|  2.9| 12.8| 17.0|
|IPS804| 194|      .7|      .0| 20.7|   .06 | 20.7|   .0|  1.4|  7.0| 12.3| 10.3|
|TMP812| 1AB|      .0|      .0|   .0|   .00 |   .0|   .0|   .0|   .0|   .0|   .0|
+------+----+--------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
+================================================================================+
```

Figure 45. RDEV IMS device statistics report

Note in the report that the SSCH and SIO counts are not comparable. The one I/O count is SSCH-based and the other is SIO-based.

The resource panels that are associated with a type of bottleneck can be established by using the feature called automatic analysis matrix.

Using the system navigator, you can go through a series of displays, beginning with the multi-interval SUMMARY and ending with some single-interval resource displays. You can back out of this tree by using the BACK command, which is assigned, by default, to PF3 and PF15. Each time that you press BACK the display pops up a level, and then you can select another line with S, D, or R.

This tree of displays is shown schematically in Figure 46 on page 69. Notice that the three types of displays are shown as both single and COMBINEd. The arrows indicate where the S, D, and R characters are appropriate.

```
    -------------                               -------------
    |  SINGLE   |-----------------------------> |  SINGLE   |
    |   GSUM    |<----------------------------  |  SUMMARY  |
    ------^------                               ------^------
          |                                           |
          |                                           |
          -------------^------------------^-----------`
                       |                  |
                       |                  |
                 ------~-----       ------~------
                 |  SINGLE   |      | SINGLE    |
                 |  DETAIL   |      | RESOURCE  |
                 ------------`      -------------

    ------------                                ------------
    |  COMBINE  |-----------------------------> |  COMBINE  |
    |   GSUM    |<----------------------------  |  SUMMARY  |
    ------^------                               ------^------
          |                                           |
          |                                           |
          ----^------------------^--------------------~-----
              |                  |                          |
              |                  |                          |
        ------~------      ------~------              ------~------
        |  COMBINE  |      | COMBINE   |              |  SINGLE   |
        |  DETAIL   |      | RESOURCE  |              |  SUMMARY  |
        -------------      -------------              ------^------
                                                            |
                                                            |
                                                   ---------~---------
                                                   |                 |
                                                   |                 |
                                             ------~------     ------~------
                                             |  SINGLE   |     |  SINGLE   |
                                             |  DETAIL   |     | RESOURCE  |
                                             -------------     ------------`
```

Figure 46. System navigator display tree

As the scheme indicates, you can go directly from the multi-interval SUMMARY (Figure 42 on page 67) to a multi-interval DETAIL and resource display. Rather than entering an *S* you can enter a *D* to produce a DETAIL display of the COMBINEd 1-hour interval from 12:00 to 13:00.

```
+---------------------------------------------------------------------------+
|                                                                           |
| Workloads:    Transaction Group 2                                         |
| Periods:      Starting 12:00 on 08/29/15 and ending 13:00 on 08/29/15     |
| Misc options: DETAIL COMBINE PLOTMIN(3)                                   |
|                                                                           |
|___________________________________________________________________________|
| WORKLOAD:     TRANSACTION GROUP 2                                         |
| PERIOD: 12:00 to 13:00 on 08/29/15                Elap =   60 M     IMSD  |
| Average of 4 intervals                                                    |
+===========================================================================+
|                            RESPONSE TIME DATA                             |
+===========================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue          |       1.25s        |       79       |       1.31      |
|Pgm Input Queue      |       0.98s        |       78       |       1.30      |
|Processing           |      10.58s        |       78       |       1.30      |
|Response Time 0      |      12.81s        |       78       |       1.30      |
|Output Queue         |       0.40s        |       76       |       1.25      |
|Response Time 1      |      12.90s        |       76       |       1.25      |
+===========================================================================+
|                              DEGRADATION DATA                             |
+===========================================================================+
|COMPETING_STATE_________TIME____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU              2.14 S   18.0|------->.   .   .   .   .   .   .   .   .|
| Using CPU in APPL   (0.94) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS    (1.20) S (10.1)|---->   .   .   .   .   .   .   .   .   .|
|Scheduling Waits       1.12 S   10.9|---->   .   .   .   .   .   .   .   .   .|
| Wait for MPP        (1.12) S (10.9)|---->   .   .   .   .   .   .   .   .   .|
RDatabase I/O Waits     7.37 S   62.1|------------============>   .   .   .   .|
| PMFG021             (7.01) S (59.0)|------------===========>.   .   .   .   .|
| BE3INDX             (0.37) S  (3.1)|->  .   .   .   .   .   .   .   .   .   .|
|z/OS Waits             0.53 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP)  (0.53) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|IMS Activity           0.53 S    4.5|->  .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL    (0.53) S  (4.5)|->  .   .   .   .   .   .   .   .   .   .|
|------------------------------------|                                   .   .|
|TOTAL (Resp Time 0)  12.81 S        |                                       |
+===========================================================================+
```

Figure 47. Multi-interval DETAIL report

As before, you can enter an *R* in the first position of the line that represents the bottleneck to be investigated. This time, you see a combined resource display for the entire hour:

```
+=========================== IMS Device Statistics ===========================+
| Period:  12:00 to 13:00 on 08/29/15                  Elap =60 M        IMSD |
+-----------------------------------------------------------------------------+
|volser|Unit|I/Os per second| %Dev| Avg q | Resp= IOSQ+ Pend+ Conn+ Disc| Open|
|      |    | Total|   IMS  | util| Length| Time| Time| Time| Time| Time| Dsns|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|PRD817| 102|    .9|      .0| 23.5|   .05 | 23.5|   .1|  1.0| 10.6| 11.8| 10.8|
|TSO802| 142|   2.2|      .0| 18.5|   .05 | 18.5|   .0|   .2|  3.4| 14.9| 14.3|
|TSO812| 14C|    .7|      .0| 23.8|   .01 | 23.8|   .0|   .2| 10.1| 13.5| 12.3|
|IPS801| 150|    .4|      .0| 15.7|   .00 | 15.7|   .0|   .0|  2.9| 12.8| 17.0|
|IPS804| 194|    .7|      .0| 20.7|   .06 | 20.7|   .0|  1.4|  7.0| 12.3| 10.3|
|TMP812| 1AB|    .0|      .0|   .0|   .00 |   .0|   .0|   .0|   .0|   .0|   .0|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
+=============================================================================+
```

Figure 48. RDEV multi-interval IMS Device Statistics report

Automatic Analysis

The system navigator helps you to explore your performance environment under manual control. EPILOG provides another facility to automate much of this process, which is known as automatic analysis.

Automation is accomplished by combining EPILOG exception-filtering capabilities with a table that you define.

The table is called an automatic-analysis matrix, or A-matrix.

Automatic analysis matrix

The automatic-analysis matrix is a table that you can use to specify the resource panels that you consider relevant to each type of bottleneck.

For example, if the main cause of poor response time is database I/O waits, you might want to see the IMS device statistics (RDEV), the IMS data set statistics (RDAS), and possibly the IMS DMB pool statistics (RDMP).

You assign each A-matrix a 1–7 character name. EPILOG prefixes the name with an at-sign (@) and keeps it as a member of the RKANPAR data set. A sample A-matrix that is called DEFAULT is provided:

```
+==============================================================================+
|   DEFAULT                                                                      |
|------------------------------------------------------------------------------|
|USING CPU      (CPA)|RCPU RSRM                                                 |
|CPU IN APPL    (APL)|RINF RSRM RCPU                                            |
|CPU IN IMS     (IMS)|RINF RSRM RCPU                                            |
|SCHDL WAITS    (SCA)|RINF RCMP RDMP RLAT ROSP RPIP RQBP RSCD RVSP              |
|SCHDL WAITS    (SCC)|RINF RCMP RDMP RLAT ROSP RPIP RQBP RSCD RVSP              |
|SCHDL WAITS    (SCE)|RINF RCMP RDMP RLAT ROSP RPIP RQBP RSCD RVSP              |
|WAIT FOR MPP   (WMP)|RINF                                                      |
|WAIT FOR RESCH (WRE)|RINF                                                      |
|WAIT FOR BMP   (WBM)|RINF                                                      |
|WAIT FOR GU    (WGU)|RINF RQBP RSCD                                            |
|INTENT CONFLT  (INC)|RINF RPIP RSCD                                            |
|SCHED BLOCKED  (SCB)|RINF RSCD                                                 |
|BLOCK MV WAIT  (BMW)|RINF RSCD                                                 |
|PSB POOL       (PSB)|RINF RPSP                                                 |
|PSBW POOL      (PSW)|RINF RCMP                                                 |
|DMB POOL       (DMB)|RINF RDMP                                                 |
|BLOCK LOADER   (BLK)|RINF RFRP                                                 |
|UNSCHEDULABLE  (UNS)|RINF                                                      |
|DB I/O WAITS   (DBW)|RINF RDEV                                                 |
|z/OS WAITS     (WMV)|RINF                                                      |
|CPU WAIT(CTL)  (CTL)|RINF RCPU                                                 |
|CP W(MPP/BMP)  (CPW)|RINF                                                      |
|PGM FETCH I/O  (PGF)|RINF RDAS RSRM                                            |
|BLDL I/O       (BLD)|RINF RDAS RSRM                                            |
|APPL I/O       (API)|RINF RDEV RSRM                                            |
|SWAPPING IN    (SWP)|RINF RPAG RSRM                                            |
|CTL PRVT PAGE  (CPP)|RINF RPAG RSRM                                            |
|DEP EXTP PAGE  (DEP)|RINF RPAG RSRM                                            |
|DEP PRVT PAGE  (DPP)|RINF RPAG RSRM                                            |
|DEP EXTP PAGE  (DEP)|RINF RPAG RSRM                                            |
|CSA PAGE       (CSA)|RINF RPAG RSRM                                            |
|EXTCSA PAGE    (ECS)|RINF RPAG RSRM                                            |
|LPA PAGE       (LPA)|RINF RPAG RSRM                                            |
|EXTLPA PAGE    (ELP)|RINF RPAG RSRM                                            |
|CROSS MEM PAG  (XMP)|RINF RPAG RSRM                                            |
|CTL PRVT(GFA)  (GPP)|RINF RPAG RSRM                                            |
|DEP PRVT(GFA)  (GDP)|RINF RPAG                                                 |
|CSA PAGE(GFA)  (GCS)|RINF RPAG                                                 |
|LPA PAGE(GFA)  (GLP)|RINF RPAG                                                 |
|X-MEM PAGE(GF  (GXM)|RINF RPAG                                                 |
+==============================================================================+
```

Figure 49. Default A-matrix table

```
+==============================================================================+
| DEFAULT                                                                      |
|------------------------------------------------------------------------------|
|IMS WAITS      (WIM)|RINF                                                     |
|MFS LOAD       (MFS)|RINF RFRP RMFP                                           |
|SPA I/O        (SPA)|RINF RDAS RSAP                                           |
|QBLKS I/O      (QBL)|RINF RDAS RQBP                                           |
|SHMSG I/O      (SHM)|RINF RDAS RQBP                                           |
|LGMSG I/O      (LMS)|RINF RDAS RQBP                                           |
|PI WAIT        (PIW)|RINF RPIP                                                |
|SYNC PNT WAIT  (SPW)|RINF                                                     |
|OTH DL/I IWT   (DLI)|RINF                                                     |
|IWAIT IN DISP  (IWD)|RINF                                                     |
|IWAIT IN TERM  (IWT)|RINF                                                     |
|LOGL LATCH     (LGL)|RINF RLAT                                                |
|SMGT LATCH     (SML)|RINF RLAT                                                |
|XCNQ LATCH     (XCL)|RINF RLAT                                                |
|DLOG LATCH     (DLL)|RINF RLAT                                                |
|IENQ LATCH     (INQ)|RINF RLAT                                                |
|HDSM LATCH     (HDS)|RINF RLAT                                                |
|QBUF LATCH     (QBF)|RINF RLAT                                                |
|OBMF LATCH     (OBF)|RINF RLAT                                                |
|VBFM LATCH     (VBF)|RINF RLAT                                                |
|DBBP LATCH     (DBB)|RINF RLAT                                                |
|DMBE LATCH     (DME)|RINF RLAT                                                |
|CBTS LATCH     (CBT)|RINF RLAT                                                |
|NQDQ LATCH     (NDQ)|RINF RLAT                                                |
|SNDQ LATCH     (SND)|RINF RLAT                                                |
|GCMD LATCH     (GCL)|RINF RLAT                                                |
|AUTH LATCH     (AUL)|RINF RLAT                                                |
|ACTL LATCH     (ACL)|RINF RLAT                                                |
|DBLK LATCH     (DBL)|RINF RLAT                                                |
|GENER LATCH    (GNL)|RINF RLAT                                                |
|OTHER LATCH    (OTL)|RINF RLAT                                                |
|ISWTCH TO CTL  (ISC)|RINF                                                     |
|ISWTCH TO LSO  (ISL)|RINF                                                     |
|OUTPUT WAITS   (OUT)|RINF                                                     |
+==============================================================================+
```

Figure 50. Default A-matrix table continued

```
+================================================================================+
| DEFAULT                                                                        |
|--------------------------------------------------------------------------------|
|IMS WAITS      (WIM)|RINF                                                       |
|MFS LOAD       (MFS)|RINF RFRP RMFP                                             |
|SPA I/O        (SPA)|RINF RDAS RSAP                                             |
|QBLKS I/O      (QBL)|RINF RDAS RQBP                                             |
|SHMSG I/O      (SHM)|RINF RDAS RQBP                                             |
|LGMSG I/O      (LMS)|RINF RDAS RQBP                                             |
|PI WAIT        (PIW)|RINF RPIP                                                  |
|SYNC PNT WAIT  (SPW)|RINF                                                       |
|OTH DL/I IWT   (DLI)|RINF                                                       |
|IWAIT IN DISP  (IWD)|RINF                                                       |
|IWAIT IN TERM  (IWT)|RINF                                                       |
|LOGL LATCH     (LGL)|RINF RLAT                                                  |
|SMGT LATCH     (SML)|RINF RLAT                                                  |
|XCNQ LATCH     (XCL)|RINF RLAT                                                  |
|DLOG LATCH     (DLL)|RINF RLAT                                                  |
|IENQ LATCH     (INQ)|RINF RLAT                                                  |
|HDSM LATCH     (HDS)|RINF RLAT                                                  |
|QBUF LATCH     (QBF)|RINF RLAT                                                  |
|OBMF LATCH     (OBF)|RINF RLAT                                                  |
|VBFM LATCH     (VBF)|RINF RLAT                                                  |
|DBBP LATCH     (DBB)|RINF RLAT                                                  |
|DMBE LATCH     (DME)|RINF RLAT                                                  |
|CBTS LATCH     (CBT)|RINF RLAT                                                  |
|NQDQ LATCH     (NDQ)|RINF RLAT                                                  |
|SNDQ LATCH     (SND)|RINF RLAT                                                  |
|GCMD LATCH     (GCL)|RINF RLAT                                                  |
|AUTH LATCH     (AUL)|RINF RLAT                                                  |
|ACTL LATCH     (ACL)|RINF RLAT                                                  |
|DBLK LATCH     (DBL)|RINF RLAT                                                  |
|GENER LATCH    (GNL)|RINF RLAT                                                  |
|OTHER LATCH    (OTL)|RINF RLAT                                                  |
|ISWTCH TO CTL  (ISC)|RINF                                                       |
|ISWTCH TO LSO  (ISL)|RINF                                                       |
|OUTPUT WAITS   (OUT)|RINF                                                       |
+================================================================================+
```

Figure 51. Default A-matrix table continued

Each row represents a bottleneck or wait reason and each is prefixed by a three-character abbreviation. While the number of reasons (rows) remains constant, the number of columns (resource panel names) changes.

For each wait reason, you can choose to associate from none to all of the available resource panels. Their order of entry in this chart indicates panel display order. When you are designing the A-matrix, it is important to keep in mind which resource panels actually apply to the area you are investigating. You create, modify, and display the A-matrix by using a set of utility commands as described in "A-matrix utilities" on page 75.

Automatic resource reports (AUTO keyword)

You can use A-matrices in various ways. If batch mode is run, the AUTO keyword substitutes the system navigator. The A-matrix also comes into effect when you use automatic resource reporting. This feature automatically generates the resource panels that are associated with the most significant wait reason.

For example, if you display the previous detail report and want to automatically display resource panels that are associated with the most significant wait reason, you enter AUTO to automatically display the resource panels. This is essentially the same thing that happens when you enter an R in the first space of the bottleneck line, as previously shown in Multi-interval DETAIL report.

In the following example, the most significant wait reason is database I/O waits. Assume that the resource panels for this wait reason, as specified in the A-matrix, are IMS device statistics (RDEV) and IMS data set statistics (RDAS). These panels display automatically when you enter AUTO as part of the command:

```
DIS GRP(2) SDATE(08/29/15) STIME(1200) -
EDATE(08/29/15) ETIME(1300) AUTO COMB
```

The following example of DETAIL Report shows the IMS statistics for the panels that automatically display when you specify AUTO as part of the command.

```
+==============================================================================+
|                                                                              |
| Workloads:    Transaction Group 2                                            |
| Periods:      Starting 12:00 on 08/29/15 and ending 13:00 on 08/29/15        |
| Misc options: DETAIL COMBINE AUTOMATIC                                       |
|                                                                              |
--------------------------------------------------------------------------------
| WORKLOAD:     TRANSACTION GROUP 2                                            |
| PERIOD: 12:00 to 13:00 on 08/29/15                Elap =   60 M     IMSD    |
| Average of 4 intervals                                                       |
+==============================================================================+
|                            RESPONSE TIME DATA                                |
+==============================================================================+
|RESPONSE_COMPONENT_____AVG. RSP. TIME_____TRANS. COUNT_____RATE (per. min.)  |
|Input Queue         |       1.25s       |       79        |       1.31        |
|Pgm Input Queue     |       0.98s       |       78        |       1.30        |
|Processing          |      10.58s       |       78        |       1.30        |
|Response Time 0     |      12.81s       |       78        |       1.30        |
|Output Queue        |       0.40s       |       76        |       1.25        |
|Response Time 1     |      12.90s       |       76        |       1.25        |
+==============================================================================+
|                              DEGRADATION DATA                                |
+==============================================================================+
|COMPETING_STATE________TIME____%_|0___1___2___3___4___5___6___7___8___9___0|
|Using CPU             2.14 S   18.0|------->.   .   .   .   .   .   .   .   .|
| Using CPU in APPL  (0.94) S  (7.9)|--->.   .   .   .   .   .   .   .   .   .|
| Using CPU in IMS   (1.20) S (10.1)|---->   .   .   .   .   .   .   .   .   .|
|Scheduling Waits      1.12 S   10.9|---->   .   .   .   .   .   .   .   .   .|
| Wait for MPP       (1.12) S (10.9)|---->   .   .   .   .   .   .   .   .   .|
RDatabase I/O Waits    7.37 S   62.1|------------=============>  .   .   .   .|
| PMFG021            (7.01) S (59.0)|------------===========>.   .   .   .   .|
| BE3INDX            (0.37) S  (3.1)|-> .   .   .   .   .   .   .   .   .   .|
|z/OS Waits            0.53 S    4.5|-> .   .   .   .   .   .   .   .   .   .|
| CPU Wait (MPP/BMP) (0.53) S  (4.5)|-> .   .   .   .   .   .   .   .   .   .|
|IMS Activity          0.53 S    4.5|-> .   .   .   .   .   .   .   .   .   .|
| ISWITCHED to CTL   (0.53) S  (4.5)|-> .   .   .   .   .   .   .   .   .   .|
|-----------------------------------|                                    .   .|
|TOTAL (Resp Time 0)  12.81 S       |                                    .   .|
+==============================================================================+
```

Figure 52. DETAIL Report with AUTO specified

```
+========================== IMS Device Statistics ==========================+
| Period:  12:00 to 13:00 on 08/29/15                  Elap =60 M       IMSD |
+----------------------------------------------------------------------------+
|volser|Unit|I/Os per second| %Dev| Avg q | Resp= IOSQ+ Pend+ Conn+ Disc| Open|
|      |    | Total|   IMS  | util| Length| Time| Time| Time| Time| Time| Dsns|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|PRD817| 102|    .9|      .0| 23.5|   .05 | 23.5|   .1|  1.0| 10.6| 11.8| 10.8|
|TSO802| 142|   2.2|      .0| 18.5|   .05 | 18.5|   .0|   .2|  3.4| 14.9| 14.3|
|TSO812| 14C|    .7|      .0| 23.8|   .01 | 23.8|   .0|   .2| 10.1| 13.5| 12.3|
|IPS801| 150|    .4|      .0| 15.7|   .00 | 15.7|   .0|   .0|  2.9| 12.8| 17.0|
|IPS804| 194|    .7|      .0| 20.7|   .06 | 20.7|   .0|  1.4|  7.0| 12.3| 10.3|
|TMP812| 1AB|    .0|      .0|   .0|   .00 |   .0|   .0|   .0|   .0|   .0|   .0|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
+============================================================================+
+========================= IMS Dataset Statistics ==========================+
| Period:  12:00 to 13:00 on 08/29/15                  Elap =60 M       IMSD |
+----------------------------------------------------------------------------+
|  Data  |      Description              | Unit |  volser  |   I/O   |  I/O   |
|  Set   |                               |      |          |  Count  |  Rate  |
+--------+-------------------------------+------+----------+---------+--------+
|  LGMG  |  Long Message Data Set        |  756 |  IMS130  |       3 |    .0  |
|  SHMG  |  Short Message Data Set       |  756 |  IMS130  |     158 |    .0  |
|  QBLK  |  Qblocks Data Set             |  756 |  IMS130  |       6 |    .0  |
|  RDS   |  Restart Data Set (Disk Log)  |  756 |  IMS130  |      10 |    .0  |
|  ACB   |  ACB Library                  |  756 |  IMS130  |       0 |    .0  |
|  DACB  |  ACB Library (DLS)            |  756 |  IMS130  |       0 |    .0  |
|  ACBB  |  ACB LibraryB                 |  756 |  IMS130  |       0 |    .0  |
|  DACBB |  ACB LibraryB (DLS)           |  756 |  IMS130  |       0 |    .0  |
|  MFS   |  Message Format Library       |  756 |  IMS130  |       0 |    .0  |
|  MFSB  |  Message Format LibraryB      |  756 |  IMS130  |       0 |    .0  |
|  TMFS  |  Test Message Format Library  |  756 |  IMS130  |    1140 |    .0  |
|  TMFSB |  Test Message Format LIbraryB |  756 |  IMS130  |       0 |    .0  |
|  SPA   |  Scratch Pad Area (Disk Spa)  |  756 |  IMS130  |       0 |    .0  |
|  MDS   |  Modstat Data Set             |  756 |  IMS130  |       0 |    .0  |
|  RCN1  |  Recon Dataset 1              |  154 |  IMS131  |      53 |    .0  |
|  RCN2  |  Recon Dataset 2              |  154 |  IMS131  |      42 |    .0  |
|  MDA   |  Mod BlocksA                  |  756 |  IMS130  |       0 |    .0  |
|  MDAB  |  Mod BlocksB                  |  756 |  IMS130  |       0 |    .0  |
+--------+-------------------------------+------+----------+---------+--------+
+============================================================================+
```

Figure 53. DETAIL Report with AUTO specified continued

If you further specify other exception criteria when you use AUTO, EPILOG displays the resource panels for those other criteria, even if they do not reflect the most significant bottleneck.

The AUTO keyword is only valid when you request a DETAILed degradation display.

By default, the A-matrix that is called DEFAULT selects the associated resource panels. You can override this setting by using the AMATRIX keyword plus AUTO. In the following example, EPILOG uses the A-matrix that is called MYMAT instead of DEFAULT to make its decisions:

```
DISPLAY GRP(2) TODAY AUTO AMATRIX(MYMAT)
```

A-matrix utilities

A-matrices are standard members of the RKANPAR partitioned data set. You can add, delete, or edit an A-matrix as a normal PDS member. EPILOG also provides a set of commands to create, delete, list, and replace A-matrix tables from the EPILOG reporter.

Create an A-matrix (CREATEM keyword)

The CREATEM command (also known as CMAT) creates a new automatic analysis member. The first operand, which is required, is a one- to seven-character A-matrix name. You can also enter any number of wait reasons, each followed by the resource panel names that are to be associated with it.

The syntax of the CREATEM command is as follows:

```
CMAT  A-matrix-Name wait-reason(resource-panels ...) ...
```

For example, the following command creates an A-matrix that is called SHIFT1, which displays the RCIP, RDEV, and RSAP resource panels whenever you use CPU in IMS. IMS is the most significant wait reason:

```
CMAT SHIFT1 IMS(RCIP RDEV RSAP)
```

The short form of the wait reasons (IMS, z/OS, SCC, and so on) can be found in Chapter 5, "Selecting EPILOG displays based on bottleneck factors," on page 55. The 21 resource panel names can be found in "Resource displays (Rccc keywords)" on page 32.

Delete an A-matrix (DELETEM keyword)

The DELETEM command (also known as DMAT) is used to delete an automatic-analysis member from the RKANPAR data set.

The only required operand is a 1–7 character A-matrix name. The syntax of the DELETEM command is as follows:

```
DMAT  A-matrix-Name
```

List an A-matrix (LISTM keyword)

The LISTM command (also known as LMAT) lists an automatic-analysis member from the RKANPAR data set.

The only operand, which is optional, is a 1–7 character A-matrix name. If you omit the operand, EPILOG issues the message: MSG EB030 AMATRIX NAME IS MISSING. To list the default A-matrix, you can enter LMAT DEFAULT. The syntax of the LISTM command is as follows:

```
LMAT A-matrix-Name
```

Replace an A-matrix (REPLACEM keyword)

The REPLACEM command (also known as RMAT) modifies and replaces an automatic-analysis member.

The first operand, which is required, is a 1–7 character A-matrix name. You can also enter any number of wait reasons, each followed by its resource panel name. The syntax of the REPLACEM command is as follows:

```
RMAT  A-matrix-Name wait-reason(resource-panels ...) ...
```

The following command adds the RCPU resource panel to the IMS (by using CPU in IMS) wait reason in A-matrix DAY and replaces the A-matrix entry for DAY in the RKANPAR data set:

```
 RMAT DAY IMS(RCPU)
```

Any existing entries within DAY remain unchanged (except for IMS, which is replaced by the new specification). The IDs that represent all the possible wait reasons can be found in Chapter 5, "Selecting EPILOG displays based on bottleneck factors," on page 55. The resource panel names can be found in "Resource displays (Rccc keywords)" on page 32.

Chapter 7. Bottleneck trend analysis

The following topics describe how to use the SUMMARY option of the DISPLAY= command with its keywords ON and PLOTPCT.

Summary display

You can use the SUMMARY option of the DISPLAY command to show trends in response time over a time period. Sometimes, however, you might want to plot the relative impact of a specific bottleneck reason on response time than the actual response time.

For example, if you notice that you often spend a large percentage of time waiting for MFS formats to be loaded, you might want to see if this problem is getting worse. In this case, you can examine the percentage of the life of an average transaction that is spent waiting for MFS load. With EPILOG, use the ON and PLOTPCT keywords (see Chapter 8, “Miscellaneous keywords in EPILOG reports,” on page 81).

In a normal summary display, the only bottleneck reason that is shown for a specific time interval is the one that accounted for the largest percentage of bottleneck during the interval. To display the same bottleneck reason for all intervals within the summary display, enter the ON keyword followed by the ID of the desired bottleneck reason. The following summary display example shows the output that is generated from the ON command:

```
+--------------------------------------------------------------------------------+
|                                                                                |
|    DIS SYSTEM SUMMARY SDATE(08/29/15) STIME(1544)                              |
|    EDATE(08/29/15) ETIME(1814) ON(MFSLOAD)                                     |
|                                                                                |
+--------------------------------------------------------------------------------+
 EPILOG
 SCREEN LOG LISTING                                08/29/15 16.42.21 PAGE   1
COPYRIGHT (C) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1991, 1992 CANDLE CORPORATION
+--------------------------------------------------------------------------------+
|                                                                                |
| WORKLOADS:    SYSTEM WIDE BOTTLENECK                                           |
| PERIODS:      STARTING ON 08/29/15 AT 15:44 AND ENDING ON 08/29/15 AT 16:59    |
| ON:           MFS LOAD                                                         |
| MISC OPTIONS: SUMMARY PLOTTIME SINGLE                                          |
|                                                                                |
+--------------------------------------------------------------------------------+
+================================================================================+
| SYSTEM WIDE                                                                    |
| PERIOD:  15:44 TO 16:59 ON 08/29/15                                            |
+--------------------------------------------------------------------------------+
|                         COMPETING TRANSACTIONS                                 |
+--------------------------------------------------------------------------------+
|DATE__START_END___MAIN_REASON(*)_TIME(-)|0S____0.2____0.4____0.6____0.8_____1|
|08/29 15:44 15:59 MFS LOAD        0.85 S|**---------------------------->      .|
|      15:59 16:14 MFS LOAD        1.14 S|****------------------------------->+|
|      16:14 16:29 MFS LOAD        0.82 S|****------------------------->.      .|
|      16:29 16:44 MFS LOAD        0.77 S|**********---------------->  .      .|
|      16:44 16:59 MFS LOAD        0.70 S|*******---------------->     .      .|
|---------------------------------------                                       |
|AVERAGE (RESP TIME 0)             0.86 S                                       |
+================================================================================+
```

Figure 54. Summary display example using ON keyword

If you also enter the PLOTPCT keyword, the graph shows both response time and the relative percentage of bottlenecks. The display in Figure 55 on page 78 shows that the impact on response time that is caused by waiting for MFS formats to be loaded is increasing over the time period displayed.


```
+-------------------------------------------------------------------------------+
|                                                                               |
|    DIS SYSTEM SUMMARY SDATE(08/29/15) STIME(1544)                             |
|    EDATE(08/29/15) ETIME(1814) ON(MFSLOAD) PLOTPCT                            |
|                                                                               |
+-------------------------------------------------------------------------------+
 EPILOG
 SCREEN LOG LISTING                             08/29/15   16.42.21  PAGE  1
COPYRIGHT (C) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1991, 1992 CANDLE CORPORATION
+-------------------------------------------------------------------------------+
|                                                                               |
| WORKLOADS:    SYSTEM WIDE BOTTLENECK                                          |
| PERIODS:      STARTING ON 08/29/15 AT 15:44 AND ENDING ON 08/29/15 AT 16:59   |
| ON:           MFS LOAD                                                        |
| MISC OPTIONS: SUMMARY PLOTPCT SINGLE                                          |
|                                                                               |
+-------------------------------------------------------------------------------+
+===============================================================================+
| SYSTEM WIDE                                                                   |
| PERIOD:  15:44 TO 16:59 ON 08/29/15                                           |
+-------------------------------------------------------------------------------+
|                          COMPETING TRANSACTIONS                               |
+-------------------------------------------------------------------------------+
|DATE__START_END___MAIN_REASON(*)_TIME(-)___%_|0__1__2__3__4__5__6__7__8__9__0|
|08/29 15:44 15:59 MFS LOAD        0.85 S  6.3|-> .  .  .  .  .  .  .  .  .  .|
|      15:59 16:14 MFS LOAD        1.14 S 11.8|--->  .  .  .  .  .  .  .  .  .|
|      16:14 16:29 MFS LOAD        0.82 S 13.5|----> .  .  .  .  .  .  .  .  .|
|      16:29 16:44 MFS LOAD        0.77 S 36.4|----------=> .  .  .  .  .  .  .|
|      16:44 16:59 MFS LOAD        0.70 S 28.6|-------->.  .  .  .  .  .  .  .|
|----------------------------------------                                       |
|AVERAGE (RESP TIME 0)            0.86 S                                        |
+===============================================================================+
```

Figure 55. Summary display example using ON and PLOTPCT keywords

This trending feature can be useful in determining the impact of tuning measures. For instance, if the MFS format library in the example were moved to a less active DASD volume, you might want to repeat the trending plot to see whether the percentage of time that is spent waiting for MFS load is reduced.

Table 21 on page 78 describes both the ON and PLOTPCT keywords.

Table 21. ON and PLOTPCT keywords

Keyword	Short	Description	Operand	Command
ON	ON	Generates summary display for a specific bottleneck reason (for trend analysis).	Options: any wait reason ID: ON(CPUWTCTL)	DISPLAY
PLOTPCT	PPCT	Used with SUMMARY and ON keywords, causes the plotting of bottleneck reason percentage rather than response time.		DISPLAY

ON keyword with summary display

Degradation trends can be analyzed by using the ON keyword with the bottleneck summary display. When the ON keyword is used, the wait time for the selected bottleneck reason (rather than worst wait) is graphed. You can monitor certain bottleneck reasons over extended periods of time.

The following summary display example shows waiting for MPP analysis that occurs over a two-week period. The wait for MPP time gradually increases.

```
CMD==> DISPLAY SYSTEM SUMMARY ON(WMP)
*******************************************************************************
==============================================================================+
| SYSTEM WIDE                                                                 |
| PERIOD:  05:23 TO 12:29 ON 08/29/15                                         |
+-----------------------------------------------------------------------------+
|                          COMPETING TRANSACTIONS                             |
+-----------------------------------------------------------------------------+
|DATE__START_END___MAIN_REASON(*)_TIME(-)|0S____0.2____0.4____0.6____0.8_____1|
|08/29 09:00 17:00 WAIT FOR MPP    0.79 S|******--------------------->.      .|
|08/30 09:00 17:00 WAIT FOR MPP    0.87 S|*******----------------------->    .|
|08/31 09:00 17:00 WAIT FOR MPP    0.77 S|**********----------------> .      .|
|09/01 09:00 17:00 WAIT FOR MPP    0.90 S|********----------------------->   .|
|09/02 09:00 17:00 WAIT FOR MPP    0.76 S|***********-------------->  .      .|
|09/03 09:00 17:00 WAIT FOR MPP    0.77 S|************--------------> .      .|
|09/04 09:00 17:00 WAIT FOR MPP    0.70 S|**********------------->    .      .|
|09/05 09:00 17:00 WAIT FOR MPP    0.81 S|****************------------>      .|
|09/06 09:00 17:00 WAIT FOR MPP    0.69 S|**********************-->   .      .|
|09/07 09:00 17:00 WAIT FOR MPP    0.98 S|***************************------> .|
|----------------------------------------                                     |
|AVERAGE (RESP TIME 0)             0.80 S                                     |
+=============================================================================+
```

Figure 56. Summary display with the ON keyword

PLOTPCT keyword

The PLOTPCT keyword can be used to analyze trends by bottleneck percent. When the PLOTPCT keyword is added to the previous display, the wait percent is graphed and the wait time.

The following summary display example shows how the system graphs the wait percent and the wait time.

```
CMD==> DISPLAY SYSTEM SUMMARY ON(WMP) PLOTPCT
+=============================================================================+
| SYSTEM WIDE                                                                 |
| PERIOD:  05:23 TO 12:29 ON 08/29/15                                         |
+-----------------------------------------------------------------------------+
|                          COMPETING TRANSACTIONS                             |
+-----------------------------------------------------------------------------+
|DATE__START_END___MAIN_REASON(*)_TIME(-)___%_|0__1__2__3__4__5__6__7__8__9__0|
|08/29 09:00 17:00 WAIT FOR MPP    0.79 S 35.1|---------=> .  .  .  .  .  .  .|
|08/30 09:00 17:00 WAIT FOR MPP    0.87 S 28.0|-------->.  .  .  .  .  .  .  .|
|08/31 09:00 17:00 WAIT FOR MPP    0.77 S 42.1|---------===>  .  .  .  .  .  .|
|09/01 09:00 17:00 WAIT FOR MPP    0.90 S 50.0|---------======>  .  .  .  .  .|
|09/02 09:00 17:00 WAIT FOR MPP    0.76 S 47.5|---------=====>.  .  .  .  .  .|
|09/03 09:00 17:00 WAIT FOR MPP    0.77 S 61.3|---------=========>  .  .  .  .|
|09/04 09:00 17:00 WAIT FOR MPP    0.70 S 58.7|---------========>.  .  .  .  .|
|09/05 09:00 17:00 WAIT FOR MPP    0.81 S 52.8|---------=======> .  .  .  .  .|
|09/06 09:00 17:00 WAIT FOR MPP    0.69 S 75.3|---------=============> .  .  .|
|09/07 09:00 17:00 WAIT FOR MPP    0.98 S 65.7|---------==========> .  .  .  .|
|----------------------------------------                                     |
|AVERAGE (RESP TIME 0)             0.53 S                                     |
+=============================================================================+
```

Figure 57. Summary display with the PLOTPCT keyword

Chapter 8. Miscellaneous keywords in EPILOG reports

Use the keywords in the following table to further qualify and define an EPILOG report.

Table 22. Miscellaneous option keywords

Keyword	Short	Description	Operand	Command
AMATRIX		Specifies automatic-analysis matrix for AUTO option; default is DEFAULT (applies only to DETAIL and SUMMARY reports).	1–7 chars	DISPLAY SET
AUTOMATIC	AUTO	Displays the resource panels that are indicated by AMATRIX for the highest wait reason (applies only to DETAIL reports).		DISPLAY SET
AVERAGE	AVG	Shows the average bottleneck time for a specified workload.		DISPLAY
BLOCK		Supplies a report title in 5 x 7 block letters.	1–12 chars	PAGESEP
CATEGORY	CAT	Limits the display to the wait-reason categories indicated by this option.	See Note 1	DISPLAY
COLOR		Turns on the extended color support that sets the 8-byte color mask.	OFF or color mask	CONTROL
COMBINE	CMB	Combine intervals. An optional multi-interval length can be provided in units of minutes or hours (default is SINGLE).	*nn*M or *nn*H	DISPLAY
COMPETING	CMP	Analyzes COMPETING transactions (see HELP STATE). Applicable to DISPLAY DETAIL and SUMMARY.		DISPLAY
DUPLICATE	DUP	Produces the specified number of separator pages (default = 1).	1–99	PAGESEP
FOLDOFF		Specifies CNT LOG does not fold output to uppercase.		CONTROL
FOLDON		Specifies CNT LOG folds output to uppercase.		CONTROL
HILITE		Turns extended color support on and sets the 8-byte highlight mask.	highlight mask	CONTROL
HOLD		Changes the display mode; holds the topmost display in the screen.		CONTROL
LIMIT	LIM	Limits the number of records that match the selection criteria. Zero means no limit (default = 10 for batch, 0 for fullscreen).	0–32767	DISPLAY SET

Keyword	Short	Description	Operand	Command
LOG		Logs the current display to the EPILOG data set (PF9/PF21).	ON OFF	CONTROL
MAXSCALE	MAXS	Allows you to supply your own values for the scaling header.	A time value; see note 2	DISPLAY
MODE		Advances the display mode (from PAGE to ROLL to HOLD to ONE and back to PAGE) (PF2/PF14).		CONTROL
ON	ON	Generates SUMMARY display for a specific BOTTLENECK reason (for trend analysis).	See Note 3	DISPLAY
ONE		Changes the display mode; allows only one display on the screen at a time.		CONTROL
PAGE		Changes the display mode; allows as many displays on the screen as can be accommodated.		CONTROL
PLOTMIN	PMIN	Plots the percentage threshold (default = 5).	0–99	DISPLAY SET
PLOTPCT	PPCT	Used with SUMMARY and ON keywords, causes the plotting of bottleneck reason percentage rather than response time.		DISPLAY
PLOTTIME	PTME	Used with SUMMARY and ON keywords, causes the plotting of bottleneck reason time.		DISPLAY
RECALL		Recalls the previous command to the input line (PF4/PF16).		CONTROL
ROLL		Changes the display mode; discards the topmost display each time you press Enter, and rolls up the displays beneath it.)		CONTROL
SCROLL		Scrolls up (PF7/PF19) or down (PF7/PF19) *n* displays or screens; *n* is optional. If you omit *n*, EPILOG assumes 1. If you enter **M**, the display scrolls up or down the maximum value.	UP *n* DOWN *n*	CONTROL
SHORT		Indicates that the title block is in condensed format.		DISPLAY INQ SET
SINGLE	SNG	Prevents a COMBINE of data; displays individual intervals, jobs, or jobsteps (SINGLE is the default).		DISPLAY

Table 22. Miscellaneous option keywords (continued)

Table 22. Miscellaneous option keywords (continued)

Keyword	Short	Description	Operand	Command
SYSID		Produces the SMF system ID (default is to accept any SMF ID).	1–4 chars	DISPLAY LOAD SET
TITLE		Generates a single-spaced report title that is enclosed in quotation marks. More than one title can be entered.	1–72 chars	DISPLAY INQ PAGESEP SET
TITLE2		Generates a double-spaced report title that is enclosed in quotation marks. More than one title can be entered.	1–72 chars	DISPLAY INQ PAGESEP SET
TITLE3		Generates a triple-spaced report title that is enclosed in quotation marks. More than one title can be entered.	1–72 chars	DISPLAY INQ PAGESEP SET
TOTAL	TOT	Shows the total bottleneck time for a specified workload.		DISPLAY

Notes:

1. Options are: DBS (database I/O),IMS (IMS waits), z/OS (z/OS waits), CPU (CPU waits), ESS (External subsystem waits), SCH (scheduling waits), OTH (others).
2. For MAXSCALE, values greater than 1 second are rounded up to the next higher multiple of 5.
3. Option: any wait reason ID: ON(CPUWTCTL).

Chapter 9. EPILOG batch reports

EPILOG reports can help you manage the performance and capacity of the data center. Three standard report jobs are included with OMEGAMON for IMS EPILOG historical component. You can modify these jobs to suit the specific requirements of your installation.

You can allow standard daily or weekly reports on the capacity, performance, or service level of your data center. These jobs are included in *thilev*.TKANSAM as members KEIJCTRS, KEIJDBAS, and KEIJSPGS.

The following topics describe one of three sample report jobs that are provided on the distribution tape. You might want to run these jobs on your system and review the output to decide which jobs apply to your installation and, which to modify or eliminate.

Data center administration reports (KEIJCTRS job)

Data center administration reports are useful if you are responsible for the performance and tuning of CPU activity. Run these reports daily to track CPU activity. It is recommended that you run these reports daily.

IMS RESPONSE TIME ANALYSIS

CTR01-1
Analysis of IMS response-time trends.

CTR01-2
Analysis of IMS response time during the prime hours of 1:00 PM to 3:00 PM.

CTR01-3
Analysis of transaction response time that exceeds a threshold of 3 seconds during prime shift.

IMS I/O ACTIVITY BOTTLENECK ANALYSIS

CTR02-1
Analysis of transactions that are impacted by spending over 20% of their time in database I/O activity.

CTR02-2
Analysis of transactions that are impacted by spending over 20% of their time in application I/O activity.

IMS CPU UTILIZATION BOTTLENECK ANALYSIS

CTR03-1
Trend analysis of IMS transactions that are delayed by IMS processor usage.

CTR03-2
Analysis of transactions that are delayed by spending over 16% of their time in processor activity.

IMS RESOURCE ANALYSIS

CTR04-1
Analysis of IMS system-wide transaction delays that are caused by DL/I IWAITS.

CTR04-2
Analysis of IMS transaction delays that are caused by IMS dispatcher IWAITS exceeding a threshold of 17%.

CTR04-3
Analysis of IMS transaction delays that are caused by more than 12% of transaction time in IMS ISWITCH to CTL.

CTR04-4
Analysis of IMS fast path transaction rates and fast path region active information.

Database administration reports (KEIJDBAS Job)

Database administration reports are useful if you are responsible for the performance and tuning of database activity. Run these reports daily.

IMS RESPONSE TIME ANALYSIS

DBA01-1
Analysis of IMS response time trends.

DBA01-2
Analysis of IMS response time during the prime hours of 1:00 PM to 3:00 PM.

DBA01-3
Analysis of IMS response time that exceeds a threshold of 3 seconds during prime shift.

IMS I/O ACTIVITY BOTTLENECK ANALYSIS

DBA02-1
Long message data set I/O bottleneck analysis.

DBA02-2
Database I/O bottleneck analysis.

DBA02-3
Short message data set I/O bottleneck analysis.

DBA02-4
Queue blocks data set I/O bottleneck analysis.

IMS SCHEDULING BOTTLENECK ANALYSIS

DBA03-1
Scheduling bottleneck analysis for competing transactions during prime time.

DBA03-2
PSB pool bottleneck analysis.

DBA03-3
DMB pool bottleneck analysis.

DBA03-4
Block-loader bottleneck analysis.

DBA03-5
Analysis of transactions degraded by WAIT for MPP.

IMS RESOURCE ANALYSIS

DBA04-1
DMB pool usage analysis.

DBA04-2
PSB pool usage analysis.

DBA04-3
Work pools usage analysis.

DBA04-4
QBUF pool usage analysis.

DBA04-5
Message format buffer pool usage analysis.

DBA04-6
Logging statistics.

DBA04-7
OSAM/ISAM buffer pool usage.

DBA04-8
IMS data set statistics.

DBA04-9
Unschedulable transactions statistics.

DBA04-10
Database error statistics.

DBA04-11
Fast path region statistics.

Systems programming department reports (KEIJSPGS job)

Systems programming department reports are useful if you are responsible for capacity management, performance, and tuning of the system. Run these reports daily.

IMS RESPONSE TIME ANALYSIS

SPG01-1
Analysis of IMS response-time trends.

SPG01-2
Analysis of IMS response time during prime time of 1:00 PM to 3:00 PM.

SPG01-3
Analysis of IMS response time that exceeds 3 seconds during prime shift—the first five occurrences are displayed.

CONTROL REGION WAIT FOR CPU ANALYSIS

SPG02-1
Control region WAIT for processor analysis.

SPG02-2
Group 1 transactions response time degradation by shortage of processor.

IMS CONTROL REGION PAGING ANALYSIS

SPG03-1
Control region private area page-in.

SPG03-2
Control region extended private area page-in.

IMS SCHEDULING ANALYSIS

SPG04-1
Degradation by all the scheduling reasons.

SPG04-2
Group 1 transactions intent conflict analysis.

SPG04-3
Group 1 transactions PSB pool analysis.

BOTTLENECK ANALYSIS OF IMS SERVICES

SPG05-1
Degradation by all the IMS services.

SPG05-2
Impact of message format services.

SPG05-3
Analysis of the I/O to short message data set.

SPG05-4
Analysis of syncpoint waits.

SPG05-5
Analysis of IWAITs in the IMS dispatcher.

SPG05-6
Analysis of the I/O to QBLKs data set.

IMS RESOURCE ANALYSIS

SPG06-1
DMB pool usage analysis.

SPG06-2
PSB pool usage analysis.

SPG06-3
Work pools usage analysis.

SPG06-4
QBUF pool usage analysis.

SPG06-5
Message format buffer pool usage analysis.

SPG06-6
Logging statistics.

SPG06-7
OSAM/ISAM buffer pool usage.

SPG06-8
IMS data set statistics.

SPG06-9
Unschedulable transactions statistics.

SPG06-10
Database error statistics.

SPG06-11
Fast path region statistics.

Chapter 10. EPILOG data store (EDS) maintenance

EPILOG provides three maintenance processors that complete useful EPILOG datastore housekeeping functions.

KEBINIT
: Writes initialization records into a newly defined VSAM cluster.

KEBMAINT
: Restores old records from a sequential SMF file or purges unwanted data from the EPILOG datastore. (When you copy from SMF, the database can be new.)

KEBUTIL
: Re-creates the EPILOG datastore from a backup data set created with IDCAMS REPRO. KEBUTIL selects specified ranges of dates and times for exclusion during reloading. You can reorganize the VSAM cluster and at the same time remove old or unwanted data.

The EPILOG datastore uses standard VSAM data set sharing and is defined with SHAREOPTION(2,3) in the KEIDFNVS job. VSAM data set sharing allows multiple readers of the EPILOG datastore and one writer in a single z/OS system. The single writer can be either the EPILOG collector or any one of the maintenance processors.

Tou must stop the EPILOG collector to run KEBINIT, KEBMAINT, or KEBUTIL. Take extra care in a multiple-system environment; EPILOG offers no protection against running maintenance processors against an EPILOG datastore on system A while an EPILOG collector is writing into it from system B.

If the EPILOG datastore is ever rendered out of use, however, it is relatively simple to re-create it from SMF by using the KEBMAINT LOAD function.

EPILOG data store (EDS) initialization processor

The KEBINIT processor initializes a newly defined VSAM cluster for use as an EPILOG datastore, and writes a single initialization record. This requirement is imposed by VSAM. The EPILOG collector cannot use the cluster in direct-access mode until it contains at least one record.

If the cluster is suballocated and you define it as REUSE, KEBINIT also empties the cluster of any old records. UNIQUE clusters cannot be cleared in this way; you must delete and redefine them.

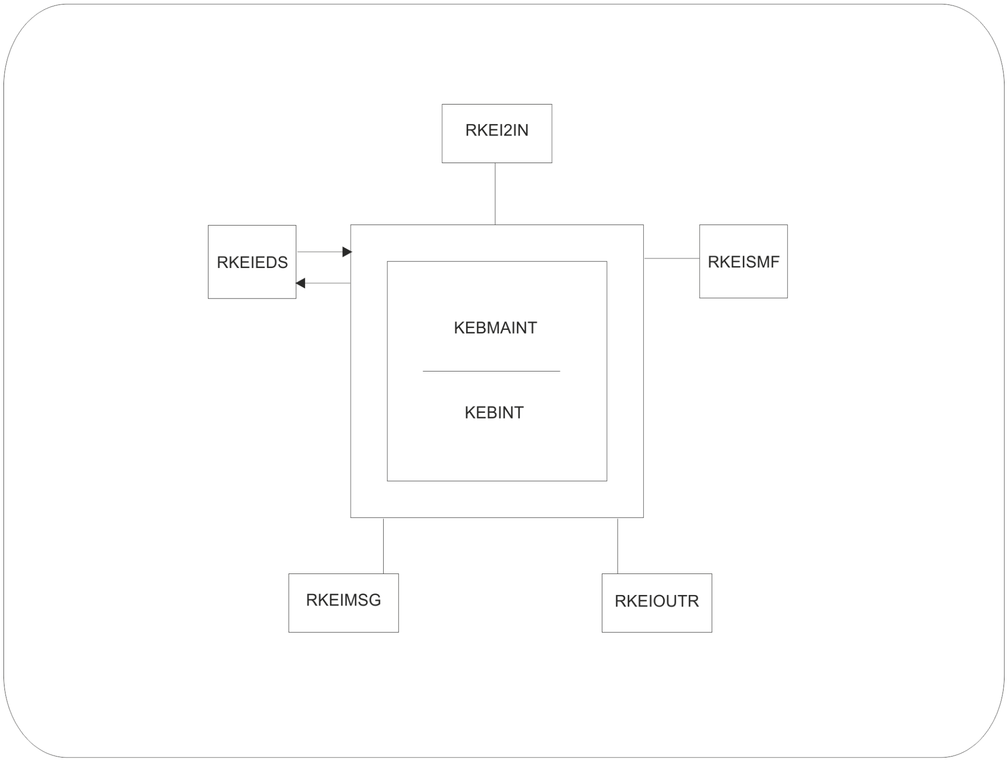

Figure 58. KEBMAINT/KEBINIT data set diagram

EPILOG data store (EDS) maintenance processor

Use the KEBMAINT processor to purge unwanted data from the EPILOG datastore and restore old records from a sequential SMF file.

Because you can use this processor to write or update the VSAM cluster, you must shut down the EPILOG collector before you run it. The data sets *rhilev.midlev*.RKANSAM(KEIDBPRG) and (KEIDBLOD) contain sample JCL to execute KEBMAINT in batch.

The following commands start KEBMAINT's prime functions:

LOAD
Loads SMF records to the EPILOG datastore.

PURGE
Purges records from the EPILOG datastore.

You can also use KEBMAINT to enter the same basic DISPLAY commands as EPILOG, although it does not support fullscreen mode. In TSO mode, KEBMAINT processes output like a batch reporter.

Load the EPILOG data store (EDS) from SMF (LOAD command)

The KEBMAINT LOAD command reads SMF records from a sequential (non-VSAM) data set and adds the records to the EPILOG datastore.

The IFASMFDP utility that is provided by IBM creates the SMF data. An RKEISMF DD statement must point to the SMF data.

The following JCL is similar to the JCL in *thilev*.TKANSAM(KEIDBLOD).

```
//EPLOAD JOB ...
//*
//*        EPLOAD    -  SAMPLE JOB TO LOAD AN EPILOG DATABASE FROM SMF
//*
//*        REMEMBER TO CHANGE THE FOLLOWING STRINGS TO THE APPROPRIATE
//*        VALUES AS DESCRIBED BELOW:
//*
//*        -RHILEV-
//*             HIGH-LEVEL INDEX OF THE RUNTIME DATASETS
//*
//*        -RVHILEV-
//*             HIGH-LEVEL INDEX OF THE RUNTIME VSAM DATASETS
//*
//*        -IMSID-
//*             ADDRESS-SPACE-SPECIFIC RUNTIME MIDDLE-LEVEL
//*             QUALIFIER OF RUNTIME DATASETS (VSAM AND NON-VSAM)
//*
//*        NOTE THAT THIS JOB WANTS THE VSAM CLUSTER “EXCLUSIVELY”
//*        (DISP=OLD); THIS IS BECAUSE KEBMAINT WILL NEED
//*        TO WRITE TO THE DATASET.
//*
//SMF    EXEC PGM=IFASMFDP
//*
//* CHANGE THE ‘SMFDATA’ DD STATEMENT BELOW FOR YOUR PURPOSES.  IT
//* SHOULD IDENTIFY A SEQUENTIAL DATASET CONTAINING THE DESIRED SMF
//* DATA
//*
//SMFDATA  DD   DSN=SMF.-IMSID-.FEB92,
//         UNIT=TAPE,VOL=SER=SMFFEB,DISP=(OLD,KEEP)
//RKEISMF  DD DSN=-MIDLEV-.-IMSID-.EISMF.DATA,
//         DISP=(,CATLG),VOL=SER=VVVVVV,UNIT=SYSDA,SPACE=(CYL,10)
//SYSPRINT DD SYSOUT=*
//SYSIN  DD  *
    INDD(SMFDATA,OPTIONS(DUMP))
    OUTDD(RKEISMF,TYPE(17))
    DATE(97037,97037)
    START(0000)
    END(2400)
```

Figure 59. KEBMAINT LOAD command

The sample job first uses IFASMFDP to extract the SMF records that KEBMAINT is interested in—record type 199 on February 6. KEBMAINT can look through all of the SMF data for February, but it is more efficient to trim the volume down first by using IFASMFDP.

KEBMAINT then LOADs degradation data for group 2 and all the resource records for the day. If you want all the EPILOG data (that is, all transaction groups and resource data) you can enter the following command:

```
LOAD SDATE(1/2/97) EDATE(1/2/97)
```

Normally you use the LOAD command to specify which types of data to include in the EPILOG datastore; adding the keyword EXCLUDE reverses the meaning of LOAD; all data is LOADed except the data that is indicated by the LOAD command.

By default KEBMAINT expects to find EPILOG records under SMF record ID 199. You can override this default with the SMF keyword:

```
LOAD GRP(2) SMF(195) SDATE(1/2/97) EDATE(1/2/97)
```

With multiple LOAD commands you can specify all the different types of data that is to LOAD. The actual LOAD processing does not occur until KEBMAINT finds an END command or end-of-file on RKEI2IN.

LOAD keywords

LOAD accepts the following keywords.

Workload
GROUP, SYSTEM, OTHER

Resource record type
: All resource record types, such as RDEV and RBUF. See "Resource displays (Rccc keywords)" on page 32 for a complete list of resource record types.

Date/time
: BAND/RANGE, DAYOFWK, ENDDATE, ENDTIME, LASTMONTH, LASTWEEK, STARTDATE, STARTTIME, THISMONTH, THISWEEK, TODAY, YESTERDAY

Miscellaneous keywords
: EXCLUDE SMF, SYSID

Note: You must specify SYSID and SMF keywords once only for all LOAD commands. The loading process uses the last occurrence of either parameter.

See "Exclude date and time (EXCLUDE command)" on page 96 for information about the EXCLUDE keyword.

For definitions of the workload, resource record type, and date/time keywords, see "Workloads" on page 34, "Resource displays (Rccc keywords)" on page 32, and "Time specifications" on page 36.

Delete records from the EDS (PURGE command)

The PURGE command deletes records (for a specific type of workload or resource panel type) from the EPILOG datastore. PURGE typically deletes all data records before a certain date, which you can use to reorganize the EPILOG datastore to make more room for current data.

Note: EPILOG collection cannot reuse purged VSAM space until you use IDCAMS REPRO to unload, re-create, and reload the EPILOG datastore.

The PURGE command can take a long time to work when you delete many records. It is far more efficient to REPRO the cluster to a non-VSAM sequential data set and then use the KEBUTIL utility to copy only the records you want back into an empty cluster (see "EPILOG data store (EDS) utility processor" on page 95). This method uses less I/O and processor resources, but you can exclude records only by date and time, not by record type.

The JCL in Figure 60 on page 93 is similar to the JCL in *rhilev.midlev*.RKANSAM(KEIDBPRG).

```
//KEBMAINT JOB ...
//*
//*        KEBMAINT  -  SAMPLE JOB TO RUN KEBMAINT AGAINST THE EPILOG
//*                     VSAM CLUSTER (EDS).
//*
//*        BEFORE RUNNING THIS JOB THE FOLLOWING PARAMETERS MUST BE
//*        CUSTOMIZED FOR YOUR INSTALLATION:
//*
//*        -RHILEV-
//*             HIGH-LEVEL INDEX OF RUNTIME DATASETS
//*
//*        -RVHILEV-
//*             HIGH-LEVEL INDEX OF RUNTIME VSAM DATASETS
//*
//*        -IMSID-
//*             ADDRESS-SPACE-SPECIFIC RUNTIME MIDDLE-LEVEL
//*             QUALIFIER OF RUNTIME DATASETS (VSAM AND NON-VSAM)
//*
//*        -RVVOL-
//*             VOLSER WHERE RUNTIME VSAM DATASET IS TO RESIDE
//*
//*        -UNIT-
//*             DASD UNIT TYPE FOR OUTPUT SEQUENTIAL EDS
//*
//*        “UNIQUE” THIS KEYWORD MUST BE CHANGED TO “REUSE” IF THE
//*                 VSAM CLUSTER IS TO BE “SUBALLOCATED”.
//*
//*
//*        NOTE THAT THIS JOB WANTS THE VSAM CLUSTER “EXCLUSIVELY”
//*        (DISP=OLD); THIS IS BECAUSE KEBMAINT WILL PROBABLY NEED
//*        TO WRITE TO THE DATASET.
//*
//*        KEBMAINT IS GENERALLY USED FOR ONE OF THE TWO FOLLOWING
//*        PURPOSES:
//*
//*        LOAD     COPY DATA INTO THE DATABASE FROM AN UNLOADED SMF
//*                 DATASET. (REQUIRES “RKEISMF” DDCARD BE SUPPLIED.)
//*        PURGE    DELETE SETS OF RECORDS FROM THE DATABASE
//*
//*        THE SAMPLE HERE IS USED TO PURGE ALL DATA FROM THE EDS WHICH
//*        HAS BEEN COLLECTED BEFORE FEBRUARY 19 OF 1995.
//*
//MAINT    EXEC PGM=KEBMAINT,REGION=2048K,PARM=’EPPROD=EI’
//*
//* EPILOG LOADLIB
//STEPLIB  DD  DISP=SHR,
//            DSN=-RHILEV-.-MIDLEV-.RKANMOD
//*
//* EPILOG EDS VSAM CLUSTER
//RKEIEDS  DD  DISP=OLD,
//            DSN=-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS
//*
//* EPILOG PARMLIBS
//RKANPAR  DD  DISP=SHR,
//            DSN=-RHILEV-.-MIDLEV-.RKANPAR
//*
//* SEQUENTIAL SMF FILE
//*RKEISMF   DD  DISP=SHR,
//             DSN=-RHILEV-.-IMSID-.EISMF.DATA
//*
//* EPILOG MESSAGES
//RKEIMSG  DD  SYSOUT=*
//*
//* EPILOG REPORT
//RKEIOUTR  DD  SYSOUT=*
//*
//* EPILOG CONTROL INPUT
//RKEI2IN  DD  *
```

Figure 60. KEBMAINT PURGE command

```
*
*          PURGE ALL DATA COLLECTED BEFORE FEBRUARY 19
*
PURGE SYS    EDATE(2/18/97)
PURGE OTH    EDATE(2/18/97)
PURGE GRP(*) EDATE(2/18/97)
PURGE RALL   EDATE(2/18/97)
PURGE GDEF   EDATE(2/18/97)
/*
//DELETE   EXEC PGM=IDCAMS,REGION=512K,COND=(0,NE)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 /*                                                     */
 /*  DELETE THE SEQUENTIAL EPILOG DATASTORE             */
 /*                                                     */
 DELETE -RHILEV-.-IMSID-.EI.SEQ NONVSAM SCRATCH PURGE
 IF LASTCC EQ 8 THEN SET MAXCC = 0
//*
//*        REORG THE EPILOG VSAM DATABASE CLUSTER
//*
//REORG    EXEC PGM=IDCAMS,REGION=512K,COND=(0,NE)
//SYSPRINT DD  SYSOUT=*
//RDS      DD  DSN=-RHILEV-.-IMSID-.EI.SEQ,DISP=(,CATLG),
//             UNIT=-UNIT-,SPACE=(CYL,(100,10),RLSE),
//             DCB=(LRECL=32600,BLKSIZE=32604,RECFM=VBS,DSORG=PS)
//SYSIN    DD  *
 /*                                                     */
 /*  COPY THE VSAM FILE TO THE SEQUENTIAL DATASET       */
 /*                                                     */
  REPRO  -
    IDS(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS) -
    OUTFILE(RDS)
  /*                                                     */
  /*  DELETE THE VSAM DATASET                            */
  /*                                                     */
  IF MAXCC = 0 THEN -
    DELETE -
      -RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS CLUSTER SCRATCH PURGE
  /*                                                     */
  /*  RE-ALLOCATE THE VSAM DATASET                       */
  /*                                                     */
  IF MAXCC = 0 THEN -
    DEFINE -
      CLUSTER( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS) -
        CYLINDERS(100 10) -
        VOL(-RVVOL-) -
        RECSZ(100 32600) -
        SPANNED -
        KEYS(32 0) -
        SHAREOPTIONS(2,3) -
        FSPC(0 0) -
        UNIQUE -
        SPEED ) -
      DATA( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS.DATA) -
        CISZ(8192) ) -
      INDEX( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS.INDEX) )
 /*                                                     */
 /*  COPY FROM THE SEQUENTIAL FILE TO THE VSAM FILE     */
 /*                                                     */
 IF MAXCC = 0 THEN -
   REPRO  -
     INFILE(RDS) -
     ODS(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS)
//
```

PURGE keywords

The following keywords work with the PURGE command.

Workload
GROUP, SYSTEM, OTHER

Resource record type
All resource record types, such as RDEV and RBUF. See "Resource displays (Rccc keywords)" on page 32 for a complete list of resource record types.

Date/time
BAND/RANGE, DAYOFWK, ENDDATE, ENDTIME LASTMONTH, LASTWEEK, NUMDAY, STARTDATE STARTTIME, THISMONTH, THISWEEK, TODAY, YESTERDAY

Either a workload or resource is required. If you do not specify either one, no records are purged and error message EB620 is displayed.

For definitions of these keywords, see "Workloads" on page 34 that describe workload keywords, "Resource displays (Rccc keywords)" on page 32 that describe resource panel types, and "Time specifications" on page 36 that describes date and time keywords.

EPILOG data store (EDS) utility processor

Any VSAM cluster such as the EPILOG datastore occasionally needs to be backed up and restored for two reasons: for archival purposes, and for reorganization of the cluster to improve performance and space usage.

When you restore data into a cluster, you can reorganize the cluster. To do this, unload it to tape (or a temporary disk data set) with IDCAMS REPRO and immediately copy it back into a new or reusable VSAM data set. Reusable means the data set is either empty (i.e., newly defined) or marked REUSE, in which case the previous contents of the cluster will be overwritten.

If the EPILOG collector writes its records to SMF, you don't need to back up the EDS. It can always be recreated from old SMF data. (However, doing a simple IDCAMS REPRO from an EDS backup tape would probably be a lot easier than recovering data from SMF, because the data may be scattered across many magtapes. For this reason many installations do backup the EDS.)

The drawback with using REPRO to perform these functions is that you must restore the entire data set. The KEBUTIL utility processor lets you specify that only *part* of the data be reloaded to the EPILOG datastore. This lets you handle your periodic backup and reorganization in one neat package.

For example, suppose you plan to keep about one month's worth of data on the EPILOG datastore. After the start of a new month, you would copy the entire EDS to a backup tape using REPRO. (See step UNLOAD in Figure 61 on page 95.) The KEBUTIL processor could then be used to copy the tape back into the EDS, *excluding* all the data from the previous month. (See step RELOAD.) If you had defined the cluster as UNIQUE it could not be marked REUSE, and so an additional step would be required to delete and redefine the cluster before using KEBUTIL to reload it. This not only gives you a backup tape of the previous month's data but also deletes the old data from the EDS and reorganizes it to reclaim the DASD space. The data set *thilev*.TKANSAM(KEIIUTIL) contains a sample job similar to this.

This method of deleting old data from the database is *much* more efficient than the KEBMAINT PURGE command; you should only use PURGE to delete small quantities of data from the EDS.

```
//...        JOB...
//UNLOAD     EXEC PGM=IDCAMS...
//RKEIVSAM   DD   DSN=-RHILEV-.-IMSID-.EI.MVS,...
.
.
.
//ARCHIVE    DD   DSN=ARCHIVE.MIDLEV.IMSID.RKEIEDS,...
  REPRO      INDD(RKEIEDS)  OUTDD(ARCHIVE)
.
.
.
//RELOAD     EXEC PGM=KEBUTIL
//UEIIN      DD   DSN=ARCHIVE.MIDLEV.IMSID.RKEIEDS,..
//RKANPAR    DD   DSN=-RHILEV-.-MIDLEV-.RKANPAR,...
//           DD   DSN=-THILEV-.TKANPAR,...

//RKEIEDS    DD   SYSOUT=A
//RKEIEDS    DD   DSN=-RHILEV-.-MIDLEV-.-IMSID-.RKEIEDS,...
//RKEI2IN    DD   * (control cmds)
```

Figure 61. KEIIUTIL job flow (skeletal JCL)

This is a typical periodic reorganization of the EPILOG datastore. The KEBUTIL processor allows only one command, the EXCLUDE command.

Exclude date and time (EXCLUDE command)

You can use the EXCLUDE command to exclude certain data based on the date and the time that the data is collected.

After the KEBUTIL EXCLUDE command is read from the RKEI2IN DDcard, it defines records for the dates and times to be excluded from the EPILOG datastore when the data is reloaded from the backup tape.

You can enter multiple EXCLUDE commands during a single run; the file is not copied until the end of the file is reached on RKEI2IN. (No data is copied unless all the EXCLUDE commands are parsed correctly.)

EXCLUDE keywords

The following are the acceptable keywords for the EXCLUDE command.

Date/time
: BAND/RANGE, DAYOFWK, ENDDATE, ENDTIME, LASTMONTH, LASTWEEK, NUMDAY STARTDATE, STARTTIME, THISMONTH, THISWEEK, TODAY, YESTERDAY

For definitions of these keywords, see "Time specifications" on page 36.

For example, to delete all the data for the previous month from the EDS, use the command **EXCLUDE LASTMONTH**.

To delete all the records from before a certain date and time, enter a command as follows:

```
EXCLUDE ENDDATE(1/02/97) ENDTIME(2400)
```

This command excludes all records written prior to midnight on 1/02/97.

When you specify start and end times with the EXCLUDE command, the BAND is the default. Thus, if the ENDTIME on the previous command was (1200), the same command would also exclude records that were written on any date up to and including the end date (1/02/97), but *only* those that were written between 00:00 hours and 12:00 hours. Records that are written between 12:01 and 24:00 would not be excluded.

To override the BAND default, you must specify the RANGE keyword. The following command excludes all records that are written up to and including 12 noon on the end date (1/02/97).

```
EXCLUDE ENDDATE(1/02/97) ENDTIME(1200) RANGE
```

If you are restoring multiple datastores that are defined as REUSE, make the following changes to the appropriate sample job:

- Add a **SKIP(1)** parameter to each **REPRO** statement to avoid copying multiple initialization records to the disk data set.
- Add a sort step after the **REPRO** statements, using the following sort statement:

  ```
  SORT FIELDS=(5,32,BI,A)
  ```

- Add a **KEBINIT** step before the **RELOAD** statement.

If your multiple data stores are not defined as REUSE, delete and reallocate the data stores after the **REPRO** step and before **KEBINIT** is run.

KEBUTIL return codes

KEBUTIL ends with a return code that notifies you of the status of the reload operation.

0
: The program completes successfully. The RKEIMSG SYSOUT output shows the number of records processed.

4
: The program completes successfully. No records were deleted.

8
: There were syntax errors in the control statements; no data was copied.

16
: There was an error in the input or output DD statements; no data was copied.

20
: The data set was only partially written. An error occurred on the VSAM file.

Sample KEBUTIL job

The *rhilev.midlev*.RKANSAM data set contains a sample job stream to unload and reload a reusable EPILOG datastore. This job deletes any data from the new EDS, which is over 5 days old *or* from a weekend.

If you do not mark your VSAM cluster REUSE or defined it as UNIQUE then the job does not work as is. You must add an extra step between the UNLOAD and RELOAD steps to delete and reallocate the cluster. (See the sample job in *rhilev.midlev*.RKANSAM(KEIDFNVS) for details.)

Note: The cluster must not be initialized with KEBINIT before reloading; the initialization record from the original EPILOG datastore is restored to the file by KEBUTIL.

Figure 62. KEIIUTIL sample job

```
//KEIUTIL JOB ...
//*
//*        KEIUTIL   -  SAMPLE JOB TO ARCHIVE THE EPILOG VSAM DATASET
//*                     TO TAPE, REORGANIZE THE VSAM CLUSTER AND
//*                     LEAVE ONLY THE MOST RECENT DATA IN THE DATASET.
//*
//*        BEFORE RUNNING THIS JOB THE FOLLOWING PARAMETERS MUST BE
//*        CUSTOMIZED FOR YOUR INSTALLATION:
//*
//*        -RHILEV-
//*             HIGH-LEVEL INDEX OF RUNTIME LIBRARIES
//*
//*        -RVHILEV-
//*             HIGH-LEVEL INDEX OF RUNTIME EPILOG VSAM DATASET
//*
//*        -RVVOL-
//*             VOLUME SERIAL OF RUNTIME EPILOG VSAM DATASET
//*
//*        -IMSID-
//*             ADDRESS-SPACE-SPECIFIC RUNTIME MIDDLE-LEVEL
//*             QUALIFIER OF RUNTIME DATASETS (VSAM AND NON-VSAM)
//*
//*
//*        UNLOAD:   DUMP THE ENTIRE VSAM DATASET TO TAPE
//* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//UNLOAD  EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD  SYSOUT=*
//RKEIEDS  DD  DISP=SHR,DSN=-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS
//ARCHIVE  DD  DSN=ARCHIVE.-MIDLEV-.-IMSID-.RKEIEDS,
//         DISP=(,KEEP),UNIT=TAPE,VOL=(,RETAIN),
//         DCB=(RECFM=VBS,LRECL=32604,BLKSIZE=32608)
//SYSIN    DD   *
     REPRO  INFILE(RKEIMSG) OUTFILE(ARCHIVE)
/*
//*
//*        DEFINE:   DELETE AND REDEFINE THE VSAM CLUSTER
//*
//*        NOTE THAT IF THE CLUSTER HAS BEEN DEFINED AS “SUBALLOCATED”
//*        (THAT IS, “REUSE” RATHER THAN “UNIQUE”) THIS STEP IS NOT
//*        NECESSARY.
//*
//DEFINE   EXEC PGM=IDCAMS,REGION=4096K,COND=(0,NE)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DELETE -RVHILEV-.-MIDLEV-.IMSID-.RKEIEDS
 DEFINE -
   CLUSTER( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS) -
     CYLINDERS(100 10) -
     VOL(-RVVOL-) -
     RECSZ(100 32600) -
     SPANNED -
     KEYS(32 0) -
     SHAREOPTIONS(2,3) -
     FSPC(0 0) -
     UNIQUE -
     SPEED ) -
   DATA( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS.DATA) -
     CISZ(8192) ) -
   INDEX( NAME(-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS.INDEX) )
/*
//*
//*        RELOAD:   RELOAD THE DATASET USING KEBUTIL WITH RECENT DATA
//*
//RELOAD  EXEC PGM=KEBUTIL,REGION=4096K,COND=(0,NE),PARM='EPPROD=EI'
//STEPLIB  DD  DISP=SHR,DSN=-RHILEV-.-IMSID-.RKANMOD
//RKANPAR  DD  DISP=SHR,DSN=-RHILEV-.-IMSID-.RKANPAR
//RKEIEDS  DD  DISP=SHR,DSN=-RVHILEV-.-MIDLEV-.-IMSID-.RKEIEDS
//UEIIN    DD  DISP=OLD,DSN=*.UNLOAD.ARCHIVE,
//         VOL=REF=*.UNLOAD.ARCHIVE
//RKEIMSG  DD  SYSOUT=*
//RKEIIN DD    *
***************************************************************
*                                                             *
*      REMOVE ANY DATA  OVER 5 DAYS OLD                       *
*                                                             *
***************************************************************
 EXCLUDE NUMDAY(5)
***************************************************************
*                                                             *
*      REMOVE ANY DATA FROM THE WEEKEND                       *
*                                                             *
***************************************************************
 EXCLUDE DAY(WEEKEND)
/*
```

If your EPILOG datastore is defined UNIQUE, you must modify the stream to delete and define the EPILOG cluster.

Performance considerations

The KEBMAINT PURGE and LOAD commands read and write to the EPILOG datastore. Depending upon the volume of records that is processed and the size of the EPILOG datastore, these commands can run for a long time.

All purging and loading must be done in the batch environment. If a large amount of data is collected each day, you might want to consider a more frequent maintenance cycle. When you are processing a large volume of record deletions, it is more efficient to use REPRO and KEBUTIL rather than PURGE.

The EPILOG datastore is a VSAM key-sequenced cluster, that is, the KSDS data set, which means all the standard restrictions for VSAM files apply to the EDS. Your own environment guides your maintenance requirements, and your tuning, backup, and reorganization strategy.

Chapter 11. Reference: Sample resource panels in EPILOG displays

The following topics provide examples of resource panels that are started by using the DISPLAY command with the resource keywords.(See “Resource displays (Rccc keywords)” on page 32.)

Many resource panels look different in a DBCTL environment; an example of a DBCTL panel is shown in “Displaying general system information (RINF panel)” on page 101.

Displaying general system information (RINF panel)

You can use RINF to determine more about the IMS system in general and the environment in which it is running.

The RINF panel, which is shown in Figure 63 on page 101, gives information about the IMS system in general and the environment in which it is running for the specified period. The 15-minute elapsed time reflects the EPILOG collection interval. If the collector was synchronized with RMF, this interval is equal to the RMF interval when the data is collected. IMSD is the ID of the monitored IMS system.

```
+======================= General System Information ==========================+
| Period:  11:45 to 12:00 on 08/29/08                    Elap =15:00 M   IMSD |
+-----------------------------------------------------------------------------+
|  IMS 3.1.0       IMS Restart Date=95.144   IMS Restart Time = 07:47:55      |
|  z/OS SP3.1.3     Model=3090-600J           Mode=Native z/OS                |
|  IMS Type = DB/DC                                                           |
+=============================================================================+
```

Figure 63. RINF General System Information report

You can also use RINF to determine the environment under which you are running. In a DBCTL environment, the RINF panel looks like the panel that is shown in Figure 64 on page 101.

```
+======================= General System Information ==========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|  IMS 3.1.0       IMS Restart Date=95.144   IMS Restart Time = 07:47:55      |
|  z/OS SP3.1.0     Model=3090                Mode=Native z/OS                |
|  IMS TYPE = DBCTL                                                           |
+=============================================================================+
```

Figure 64. RINF General System Information report (DBCTL)

RINF panels for an IMS in various remote site recovery (RSR) states

The following example of a RINF General System Information report shows how the system displays IMS general system information and indicates that the status of RSR is active.

```
+----------------------- General System Information --------------------------+
| Period:  09:07 to 09:14 on 08/29/08                  Elap = 6:59 M     I51N |
+-----------------------------------------------------------------------------+
| IMS 6.1.0       IMS Restart Date=97.002   IMS Restart Time=08:47:51         |
| z/OS SP5.2.0    Model=9672                Mode=Native                       |
| IMS Type = DBCTL                          RSR Status = Active               |
+-----------------------------------------------------------------------------+
+=============================================================================+
```

Figure 65. RINF General System Information report

The following General System Information panel shows that RSR is in tracking status.


```
+----------------------- General System Information -------------------------+
| Period:  09:07 to 09:14 on 08/29/08                  Elap = 6:59 M    I51N |
+---------------------------------------------------------------------------+
| IMS 6.1.0       IMS Restart Date=97.002   IMS Restart Time=08:47:51        |
| z/OS SP5.2.0    Model=9672                Mode=Native                      |
| IMS Type = DBCTL                          RSR Status = Tracking            |
+---------------------------------------------------------------------------+
+===========================================================================+
```

Figure 66. RINF General System Information report

The following General System Information panel shows the RSR status is none.

```
+----------------------- General System Information -------------------------+
| Period:  09:07 to 09:14 on 08/29/08                  Elap = 6:59 M     I51N |
+----------------------------------------------------------------------------+
| IMS 6.1.0       IMS Restart Date=97.002   IMS Restart Time=08:47:51         |
| z/OS SP5.2.0    Model=9672                Mode=Native                       |
| IMS Type = DBCTL                          RSR Status = None                 |
+----------------------------------------------------------------------------+
+============================================================================+
```

Figure 67. RINF general system information report

Displaying CPU activity (RCPU panel)

The CPU Activity panel shows the percentage of time each region uses the CPU for the 15-minute interval that is specified as the EPILOG collection interval.

The following example of the RCPU panel shows how the system displays IMSstatistics based on the Transport Manager Subsystem (TMS) address space. Note the new TMS region type that is used if you have Remote Site Recovery (RSR).

```
+============================== CPU Activity ================================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
| Region   Type   TCB     SRB    Total    | Region   Type   TCB    SRB   Total   |
| ------   ----   ---     ---    -----    | ------   ----   ---    ---   -----   |
| CONTROL  CTL   17.2 %  4.4 %  21.6 % | MESSAGE4 MPP    5.2 %  1.2 %   6.4 % |
| MESSAGE1 MPP    3.8 %   .9 %   4.7 % | FPMSG1   FP     9.7 %  3.2 %   2.9 % |
| MESSAGE2 MPP   11.7 %  2.0 %  13.7 % | FPMSG3   FP     1.0 %   .3 %   1.3 % |
| MESSAGE3 MPP    4.5 %  1.1 %   5.6 % |                                      |
| IMS510TM TMS     .4 %   .0 %    .4 % |                                      |
+----------------------------------------------------------------------------+
+============================================================================+
```

Figure 68. RCPU panel

The following are the fields in the RCPU panel:

Region
: Region name

Type
: Type of region selected

TCB
: Percentage of time during the interval that the processor time is used by the region job step TCB

SRB
: Percentage of time during the interval that the processor time is used by the region job step SRB

Total
: TCB + SRB

Displaying SRM Statistics (RSRM panel)

The SRM Statistics panel shows information about the address spaces or regions that belong to IMS. This panel does not measure the resources but uses service units to track the computer resources a region uses.

To limit the scope of the RSRM panel, add a suboperand. For example:

DISPLAY RSRM (IMSTCTL)
Limits scope of display to the region named IMSTCTL.

DISPLAY RSRM (MESSAGE*)
Displays all regions that begin with MESSAGE.

Note the new Transport Manager Subsystem (TMS) region type, used if you have Remote Site Recovery (RSR).

This panel is an RSRM panel for an IMS that uses a TMS address space.

```
+============================ SRM Statistics =================================+
| Period:  13:00 to 13:15 on 08/29/08                    Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|           | Rgn |   CPU     |   I/O     |   MSO     |   Total   |      |  I/Os   |
|  Region   | Type| Srv Unts  | Srv Unts  | Srv Unts  | Srv Unts  | HUIC | per sec |
+-----------+-----+-----------+-----------+-----------+-----------+------+---------+
| IMSTEST   | CTL |       1M  |     813K  |        2M |        4M |  255 |     3.6 |
| DBRCT     | DBRC|      36K  |      35K  |       12K |       85K |  255 |      .1 |
| DLISAST   | DLS |     513K  |      93K  |      778K |        1M |  255 |      .6 |
+-----------+-----+-----------+-----------+-----------+-----------+------+---------+
| MESSAGE1  | MPP |     4192  |     2960  |       896 |     8048  |  157 |      .0 |
| MESSAGE2  | MPP |      98K  |      52K  |       44K |      14K  |  255 |      .3 |
| MESSAGE3  | MPP |        0  |        0  |         0 |        0  |   80 |      .0 |
| FPMSG2    | FP  |      16K  |      18K  |       18K |      36K  |   53 |      .1 |
| FPMSG3    | FP  |     536K  |     285K  |      256K |       1M  |  255 |     1.4 |
+-----------+-----+-----------+-----------+-----------+-----------+------+---------+
|IMS510TM   | TMS |      528  |       51  |         0 |      579  |  240 |      .1 |
+-----------+-----+-----------+-----------+-----------+-----------+------+---------+
| Total     |     |           |           |           |           |      |         |
| Dependent|      |     655K  |     679K  |      349K |       1M  |      |     1.9 |
+-----------+-----+-----------+-----------+-----------+-----------+------+---------+
| Total     |     |       2M  |       1M  |        3M |       7M  |      |     6.2 |
+=============================================================================+
```

Figure 69. RSRM Panel (Including region type description)

The following are the fields in the RSRM panel:

Region
Region name

Rgn Type
Type of region

CPU Srv Unts
Total CPU service units for this interval

I/O Srv Unts
Total I/O service units for this interval

MSO Srv Unts
Total main storage occupancy service units for this interval

Total Srv Unts
CPU + I/O + MSO service unit total

HUIC
Highest unreferenced interval count for this interval

I/Os per sec
I/O rate per second for this interval

Total Dependent
: Accumulation for dependent region data

Total
: Accumulation for all regions shown

Displaying Paging and Storage Information (RPAG panel)

The Paging and Storage panel shows IMS paging, swapping, and storage statistics for each region.

Note the new Transport Manager Subsystem (TMS) region type, used if you have Remote Site Recovery (RSR).

This panel is an RPAG panel for an IMS that uses a TMS address space.

```
+=========================== Paging And Storage ============================+
| Period:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+---------------------------------------------------------------------------+
|          | Rgn |  Work  |   Avg Frame Counts    |   Page/sec  | Swap page/sec |
| Region   | Type|  Set   | Real = Fixed+NonFix   |  In  |  Out |  In    |  Out |
+----------+-----+-------+------+-------+--------+------+------+-------+-------+
| IMSPROD  | CTL | 2048K |  507 |  104 |    403 |  5.6 |  4.3 |    .0 |    .0 |
+----------+-----+-------+------+-------+--------+------+------+-------|-------+
| MESSAGE1 | MPP |  376K |   94 |   20 |     74 |   .8 |      |    .0 |    .0 |
| MESSAGE2 | MPP |  360K |   90 |   19 |     71 |  1.0 |   .8 |    .0 |    .0 |
| MESSAGE3 | MPP |  344K |   86 |   22 |     64 |  1.4 |  1.2 |    .0 |    .0 |
| MESSAGE4 | MPP |  380K |   95 |   20 |     75 |  1.1 |   .7 |    .0 |    .0 |
| MESSAGE5 | MPP |  348K |   87 |   19 |     68 |  1.2 |   .3 |    .0 |    .0 |
| FPMSG2   | FP  |  572K |  143 |   27 |    116 |  3.4 |  1.0 |    .0 |    .0 |
| FPMSG4   | FP  |  408K |  102 |   21 |     81 |  2.9 |   .9 |    .0 |    .0 |
+----------+-----+-------+------+-------+--------+------+------+-------+-------+
|IMS510TM  | TMS |  516K |  129 |   19 |   110  |   .2 |   .0 |    .0 |    .0 |
+----------+-----+-------+------+-------+--------+------+------+-------+-------+
| Total    |     |       |      |       |        |      |      |       |       |
| Dependent|     | 2788K |  697 |  148 |    549 | 11.3 |  4.9 |    .0 |    .0 |
+----------+-----+-------+------+-------+--------+------+------+-------+-------+
| Total    |     | 4816K | 1204 |  252 |    952 | 16.9 |  9.2 |    .0 |    .0 |
+===========================================================================+
```

Figure 70. RPAG panel with region type description

The following are the fields in the RPAG panel.

Region
: Region name

Rgn Type
: Type of region

Work Set
: Average amount of real storage that is used for this interval (the value is expressed in Ks)

Real
: Average number of page frames that are in real storage (Fixed + NonFix)

Fixed
: Average number of page frames that are fixed in real storage

NonFix
: Average number of page frames that are not fixed in real storage

Page In
: Page-in rate per second

Page Out
: Page-out rate per second

Swap In
: Swap-in rate per second

Swap Out
: Swap-out rate per second

To limit the scope of the RPAG panel, add a suboperand. For example:

DISPLAY RPAG (IMSPROD)
Limits scope of the display to the region named IMSPROD.

DISPLAY RPAG (MESSAGE*)
Displays all regions that begin with MESSAGE.

Displaying IMS device statistics (RDEV panel)

The IMS Device Statistics panel shows information about the direct-access devices that are allocated to IMS regions.

The following example of an IMS Device Statistics report shows how the system displays IMS statistics that are based on the devices that are allocated to IMS regions.

```
+========================== IMS Device Statistics ===========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|volser|Unit|I/Os per second| %Dev| Avg q | Resp= IOSQ+ Pend+ Conn+ Disc| Open|
|      |    | Total|   IMS   | util| Length| Time| Time| Time| Time| Time| Dsns|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|PRD817| 102|    .9|      .0| 23.5|   .05 | 23.5|   .1|  1.0| 10.6| 11.8| 10.8|
|TSO802| 142|   2.2|      .0| 18.5|   .05 | 18.5|   .0|   .2|  3.4| 14.9| 14.3|
|TSO812| 14C|    .7|      .0| 23.8|   .01 | 23.8|   .0|   .2| 10.1| 13.5| 12.3|
|IPS801| 150|    .4|      .0| 15.7|   .00 | 15.7|   .0|   .0|  2.9| 12.8| 17.0|
|IPS804| 194|    .7|      .0| 20.7|   .06 | 20.7|   .0|  1.4|  7.0| 12.3| 10.3|
|TMP812| 1AB|    .0|      .0|   .0|   .00 |   .0|   .0|   .0|   .0|   .0|   .0|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|Epilog/IMS is not synchronized with PMF                                      |
+=============================================================================+
```

Figure 71. RDEV IMS Device Statistics report

The following are the fields in the RDEV panel:

volser
Volume serial ID where one or more IMS data sets are stored.

Unit
Unit name for this volser.

I/Os per sec-Total
I/Os per second for IMS and non- IMS data sets (for this unit). This number represents SSCH counts taken from the CMB.

OSAM IMS databases are maintained in separate IMS control blocks and might not be included in this value.

I/Os per sec-IMS
I/Os per second for IMS only data set activity. This number represents EXCP counts taken from the TCT.

OSAM IMS databases are maintained in separate IMS control blocks and might not be included in this value.

% Dev Util
Percentage of time during the interval when the device is in use. This time includes connect/ disconnect time (time in I/O operations) and reserved time that is not involved in I/O operations.

Avg q Length
Average number of requests that are enqueued against this device.

Resp Time
Response time in milliseconds that totals IOSQ, PEND, CONN, and DISC.

IOSQ Time
Average time during interval when SIO is enqueued against this device (in milliseconds).

Pend Time
: Average time during interval an I/O request waited for an available path to this device (in milliseconds).

Conn Time
: Average time during interval in which the path to this device is actually transferring data (in milliseconds).

Disc Time
: Average time during interval in which this device is in use but not transferring data (in milliseconds).

Open Dsns
: Number of OPEN data sets on this volume.

To limit the scope of the display, add a suboperand. For example:

DISPLAY RDEV (SYS*)
: Displays all devices that begins with SYS.

DISPLAY RDEV (14*)
: Displays all devices that begins with 14.

DISPLAY RDEV (U(C38))
: Limits scope of the display to the unit named C38.

DISPLAY RDEV (V(SYS001))
: Limits scope of the display to the volume named SYS001.

If you do not use the U and V suboperands to specify unit and volume names (as shown in the first two examples), EPILOG determines whether the suboperand designates a unit or a volser according to the following rules:

- A *volser* is a six-character volume serial ID containing a combination of numerals from 0–9 or letters from A—Z.
- A *unit* is a three-character unit name that contains a combination of numerals from 0–9 or letters from A—F.
- When a suboperand is processed, each character is checked. If the length of the suboperand is greater than three, or a character is greater than **F** (that is, not a valid hexadecimal number), it is considered a volser.

Displaying IMS data set statistics (RDAS panel)

The IMS Dataset Statistics panel shows information about the data sets that the IMS control region uses. These data sets are systems data sets and are not under user access. These data sets are on the devices that are shown in the IMS device statistics panel.

The following example of an RDAS IMS DASD data set report shows how the system displays IMS statistics that are based on the IMS control region data sets.

```
+========================== IMS Device Statistics ==========================+
| Period:  12:00 to 13:00 on 08/29/08                   Elap =60 M        IMSD |
+-----------------------------------------------------------------------------+
|volser|Unit|I/Os per second| %Dev| Avg q | Resp= IOSQ+ Pend+ Conn+ Disc| Open|
|      |    | Total|   IMS  | util| Length| Time| Time| Time| Time| Time| Dsns|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
|PRD817| 102|    .9|      .0| 23.5|    .05 | 23.5|   .1|  1.0| 10.6| 11.8| 10.8|
|TSO802| 142|   2.2|      .0| 18.5|    .05 | 18.5|   .0|   .2|  3.4| 14.9| 14.3|
|TSO812| 14C|    .7|      .0| 23.8|    .01 | 23.8|   .0|   .2| 10.1| 13.5| 12.3|
|IPS801| 150|    .4|      .0| 15.7|    .00 | 15.7|   .0|   .0|  2.9| 12.8| 17.0|
|IPS804| 194|    .7|      .0| 20.7|    .06 | 20.7|   .0|  1.4|  7.0| 12.3| 10.3|
|TMP812| 1AB|    .0|      .0|   .0|    .00 |   .0|   .0|   .0|   .0|   .0|   .0|
+------+----+------+--------+-----+-------+-----+-----+-----+-----+-----+-----+
+=============================================================================+
+========================= IMS Dataset Statistics ==========================+
| Period:  12:00 to 13:00 on 08/29/08                   Elap =60 M        IMSD |
+-----------------------------------------------------------------------------+
|  Data   |       Description               | Unit |  volser   |   I/O    | I/O     |
|  Set    |                                 |      |           |  Count   | Rate    |
+---------+---------------------------------+------+-----------+----------+---------+
|  LGMG   |  Long Message Data Set          |  756 |  IMS130   |        3 |  .0     |
|  SHMG   |  Short Message Data Set         |  756 |  IMS130   |      158 |  .0     |
|  QBLK   |  Qblocks Data Set               |  756 |  IMS130   |        6 |  .0     |
|  RDS    |  Restart Data Set (Disk Log)    |  756 |  IMS130   |       10 |  .0     |
|  ACB    |  ACB Library                    |  756 |  IMS130   |        0 |  .0     |
|  DACB   |  ACB Library (DLS)              |  756 |  IMS130   |        0 |  .0     |
|  ACBB   |  ACB LibraryB                   |  756 |  IMS130   |        0 |  .0     |
|  DACBB  |  ACB LibraryB (DLS)             |  756 |  IMS130   |        0 |  .0     |
|  MFS    |  Message Format Library         |  756 |  IMS130   |        0 |  .0     |
|  MFSB   |  Message Format LibraryB        |  756 |  IMS130   |        0 |  .0     |
|  TMFS   |  Test Message Format Library    |  756 |  IMS130   |     1140 |  .0     |
|  TMFSB  |  Test Message Format LIbraryB   |  756 |  IMS130   |        0 |  .0     |
|  SPA    |  Scratch Pad Area (Disk Spa)    |  756 |  IMS130   |        0 |  .0     |
|  MDS    |  Modstat Data Set               |  756 |  IMS130   |        0 |  .0     |
|  RCN1   |  Recon Dataset 1                |  154 |  IMS131   |       53 |  .0     |
|  RCN2   |  Recon Dataset 2                |  154 |  IMS131   |       42 |  .0     |
|  MDA    |  Mod BlocksA                    |  756 |  IMS130   |        0 |  .0     |
|  MDAB   |  Mod BlocksB                    |  756 |  IMS130   |        0 |  .0     |
+---------+---------------------------------+------+-----------+----------+---------+
+=============================================================================+
```

Figure 72. RDAS IMS DASD data set report

The RDAS panel includes the following fields:

Dataset
Abbreviated version of IMS data set name.

Description
Description of IMS data set.

Unit
Unit name on which IMS data set are stored.

volser
Volume serial ID on which IMS data set are stored.

I/O Count
The cumulative I/Os seen for the interval.

I/O Rate
The number of I/Os per second for the interval.

Enhanced I/O (RDAS panel)

Enhanced I/O support provides a second version of the RDAS report.

The following example of the RDAS IMS Data Set Statistics report shows how the system displays IMSstatistics based on the IMS data sets. A plus (+) sign in the Volume List column signifies that the data set spans more than two volumes.

```
+------------------------- IMS Dataset Statistics -----------------------------+
| PERIOD:  10:34 TO 10:44 ON 08/29/08                   ELAP = 9:47 M      I51A |
+-----------------------------------------------------------------------------|
| Jobname   | DDName    |   EXCP    |   EXCP    |          Volume List          |
|           |           |   Count   |   Rate    |                               |
+-----------+-----------+-----------+-----------+----------+--------------------|
| IMS510AC | IMSACBA   |        53 |       .00 | IMS100   IMS200  IMS300  IMS400 + |
| IMS510AD | IMSACBA   |       213 |       .00 | IMS100   IMS200  IMS300  IMS400 + |
| IMS510AC | IMSACBB   |         0 |       .00 | IMS100                          |
| IMS510AD | IMSACBB   |         0 |       .00 | IMS200                          |
| IMS510AC | IMSTFMTA |         0 |       .00 | IMS100                          |
| IMS510AC | IMSTFMTB |         0 |       .00 | IMS300                          |
| IMS510AC | LGMSG     |       129 |       .00 | IMS100   IMS200                 |
| IMS510AC | SHMSG     |        66 |       .00 | IMS100   IMS200                 |
| IMS510AC | QBLKS     |        58 |       .00 | IMS100                          |
| IMS510AC | FORMATA   |        48 |       .00 | IMS100                          |
| IMS510AC | FORMATB   |         0 |       .00 | IMS100                          |
| IMS510AB | RECON1    |       237 |       .00 | IMS100                          |
| IMS510AB | RECON2    |       223 |       .00 | IMS200                          |
| IMS510AC | IMSRDS    |        13 |       .00 | IMS100                          |
| IMS510AC | MODBLKSA |        10 |       .00 | IMS100                          |
| IMS510AC | MODBLKSB |         0 |       .00 | IMS200                          |
| IMS510AC | MATRIXA   |        12 |       .00 | IMS100                          |
| IMS510AC | MATRIXB   |         0 |       .00 | IMS300                          |
| IMS510AC | MODSTAT   |         2 |       .00 | IMS100                          |
+==============================================================================+
```

Figure 73. RDAS IMS Data Set Statistics report (second version)

The following are the fields in the RDAS panel:

Jobname
: Jobname of the IMS Region owning the IMS data set

DDName
: DDName to which the data set is allocated

EXCP Count
: Total count of EXCPs for ALL IMS data sets allocated to the DDName

EXCP Rate
: Number of EXCPs per second for ALL IMS data sets allocated to the DDName

Volume List
: List of volumes where the IMS data sets are allocated

Displaying IMS DDName statistics (RDDN panel)

Enhanced I/O Support also provides the RDDN report. Use this report with the second version of the RDAS report. The DDName of the IMS data set in which you are interested is taken from the RDAS report and provided as a parameter to the RDDN operand.

For example, use IMSACBA, from the DDName column in the RDAS panel as shown here:

```
        DISPLAY RDDN(IMSACBA)
```

This command produces the following report.

If no parameter is provided, statistics for all DDNames display.

```
+--------------------------- IMS DDName Statistics ---------------------------+
| Period:  08:50 to 08:59 on 08/29/08                  Elap = 8:19 M     I51N |
+-----------------------------------------------------------------------------+
| DDName: IMSACBA  Jobname: IMS510AC                                          |
+-----------------------------------------------------------------------------+
|   Volser  |                 Dataset Name                 |  EXCP   |  EXCP   |
|           |                                              |  Count  |  Rate   |
+-----------+----------------------------------------------+---------+---------|
|  IMS100   | IMS.V510.ACBLIBA.LIB1                        |      50 |    0.00 |
|  IMS200   | IMS.V510.ACBLIBA.LIB1                        |       2 |    0.00 |
|  IMS300   | IMS.V510.ACBLIBA.LIB2                        |       1 |    0.00 |
+-----------------------------------------------------------------------------+
| DDName: IMSACBA  Jobname: IMS510AD
+-----------------------------------------------------------------------------+
|   Volser  |                 Dataset Name                 |  EXCP   |   EXCP  |
|           |                                              |  Count  |   Rate  |
+-----------+----------------------------------------------+---------+---------|
|  IMS100   | IMS.V510.ACBLIBA.LIB1                        |     200 |    0.00 |
|  IMS200   | IMS.V510.ACBLIBA.LIB1                        |       3 |    0.00 |
|  IMS300   | IMS.V510.ACBLIBA.LIB2                        |      10 |    0.00 |
+-----------------------------------------------------------------------------+
```

Figure 74. RDDN IMS DDName Statistics report

The following are the fields in the RDDN panel:

Volser
Volume serial ID on which IMS data sets are stored.

Data set Name
Name of the data set

EXCP Count
Total count of EXCPs for ALL IMS data sets allocated to the DDName

EXCP Rate
Number of EXCPs per second for ALL IMS data sets allocated to the DDName

Displaying IMS scheduling statistics (RSCD panel)

The IMS Scheduling Statistics panel shows the number of transactions that are scheduled, how many succeeded and failed, and general problem areas. Under `Other Failures` are those transactions that are scheduled, excluding transactions that occur because of database access, program malfunction, or transaction problems.

The following example of an IMS scheduling report shows how the system displays IMS transaction statistics.

```
+========================= IMS Scheduling Statistics =========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|Scheduling|Successful |Scheduling|Program  |Priority|Intent     |Other       |
|Attempts  |Schedulings|Failures  |Conflicts|Cutoffs |Failures(*)|Failures(**)|
|----------+-----------+----------+---------+--------+-----------+------------+
|    4478  |     3287  |       5  |       1 |      0 |         0 |     1185   |
|----------+-----------+----------+---------+--------+-----------+------------+
|   (*)Includes Pool Space Failures    (**)Includes Trx, Pgm or DB Stopped    |
+=============================================================================+
```

Figure 75. RSCD IMS Scheduling Statistics report

The following are the fields in the RSCD panel:

Scheduling Attempts
The cumulative number of scheduling attempts that are seen for this interval.

Successful Schedulings
The cumulative number of scheduling attempts that are seen for this interval, which are successful.

Scheduling Failures
: The cumulative number of scheduling attempts that are seen for this interval, which are not successful.

Program Conflicts
: The cumulative number of program conflicts that are seen for this interval.

Priority Cutoffs
: The cumulative number of priority cutoffs that are seen for this interval.

Intent Failures
: The cumulative number of intent failures that are seen for this interval.

Other Failures
: The cumulative number of non-intent failures that are seen for this interval.

Displaying IMS Latch Statistics (RLAT panel)

Certain services of IMS can be used by only one task at a time. Latches are used to serialize the use of these services. If you access the component for dynamic logging, for example, no other user would be able to complete the same function.

The IMS Latch Statistics panel, which is shown in Figure 76 on page 110, specifies the types of latches, the number of times a latch is assigned exclusively or shared, and the number of IWAITS issued for the latch.

```
+========================== IMS Latch Statistics ============================+
| Period:  11:45 to 12:00 on 08/29/08                   Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
| Latch|           Usage                    | Exclusive | Shared    |  Waited   |
| Name |                                    |  Latch    |  Latch    | For Latch |
|      |                                    |  Assigned | Assigned  |           |
+------+------------------------------------+-----------+-----------+-----------+
| LOGL |  Logical Logger                    |      0    |      0    |      0    |
| SMGT |  Storage Management                |      0    |      0    |      2    |
| XCNQ |  Exclusive Control ENQ/DEQ(PI)     |      0    |      0    |      9    |
| DMSG |  HD Space Mangement                |      0    |      0    |      0    |
| QMBP |  Queue Buffer Management           |      0    |      0    |      0    |
| GCMD |  Global Command Processing         |      0    |      0    |      0    |
| AUTH |  Authorization Processing          |      0    |      0    |      0    |
| ACTL |  Statistics Logging                |      0    |      0    |      2    |
| IOBH |  ISAM/OSAM Buffer Manager          |      0    |      0    |      0    |
| VSAM |  VSAM Buffer Pool Latch            |      0    |      0    |      0    |
| DBBP |  Database Buffer Pool              |      0    |      0    |      0    |
| DMBE |  DMD Extended                      |      0    |      0    |      0    |
| CBTS |  Dynamic Control Block             |      0    |      0    |      3    |
| IENQ |  IENQ - IDEQ                       |      0    |      0    |      0    |
| SMBQ |  SMBENQ - DEQ                      |      0    |      0    |      0    |
| DBLK |  Free Dep Region Blocks            |      0    |      0    |      0    |
+------+------------------------------------+-----------+-----------+-----------+
| TOTAL|                                    |      0    |      0    |     16    |
+================================================================================+
```

Figure 76. RLAT IMS Latch Statistics report

The following are the fields in the RLAT panel:

Latch Name
: Four-character name that IMS uses when the latch is referenced.

Usage
: A description of how the latch is used.

Exclusive Latch Assigned
: Average number of times a “Getlatch” request is granted for exclusive use of the latch.

Shared Latch Assigned
: Average number of times a “Getlatch” request is granted for shared use of the latch.

Waited for Latch
Average number of times a “Getlatch” request is not granted and resulted in a wait.

Displaying IMS Logging Statistics (RLOG panel)

The IMS Logging Statistics panel provides information about the logging environment and statistics such as the number of log records that are written, write ahead requests, and OLDS buffers that are defined and available.

The following example of an RLOG IMS DASD Logging Statistics report shows how the system displays IMS statistics that are based on the logging environment.

```
+========================== IMS Logging Statistics ===========================+
| Period:  19:12 to 19:29 on 08/29/08                  Elap =17:00 M     IMSD |
+-----------------------------------------------------------------------------+
| PART I  -  LOGGING ENVIRONMENT                                              |
+----------+-------------+--------------+---------------+-----------+---------+
|  OLDS    |     OLDS    |     OLDS     |     WADS      |   WADS    |  WADS   |
| DEFINED  |    STOPPED  |   IN ERROR   |    DEFINED    |   USED    | SPARES  |
+----------+-------------+--------------+---------------+-----------+---------+
|        4 |           0 |            0 |           3   |     2     |    1    |
+----------+-------------+--------------+---------------+-----------+---------+
| OLDS LOGGING = DUAL      WADS LOGGING = DUAL      AUTO ARCHIVE LIMIT =    1 |
+-----------------------------------------------------------------------------+
+-----------------------------------------------------------------------------+
| PART II -  LOGGING STATISTICS                                               |
+----------+-------------+--------------+---------------+-----------+---------+
|  LOG     | WRITE AHEAD | DC WAITS FOR | OUTPUT BUFFER |  CHECKPOINT OUTPUT  |
| RECORDS  |  REQUESTS   | WRITE AHEAD  |     WAITS     |    BUFFER WAITS     |
+----------+-------------+--------------+---------------+-----------+---------+
|        0 |           0 |            0 |           0   |         0           |
+----------+-------------+--------------+---------------+-----------+---------+
+----------+-------------+--------------+---------------+-----------+---------+
|  WADS    | # 2K BLOCKS | OLDS BLOCKS  |  OLDS BLOCKS  |      # SYSTEM       |
| EXCPVRS  |     WADS    |  WRITTEN     |     READ      |    CHECKPOINTS      |
+----------+-------------+--------------+---------------+---------------------|
|        0 |           0 |            0 |           0   |         0           |
+----------+-------------+--------------+---------------+---------------------+
+-----------------------------------------------------------------------------+
| PART III - OLDS BUFFER STATISTICS                                           |
+----------+-------------+--------------+---------------+-----------+---------+
|  DEFINED |  AVAILABLE  |   USED FOR   |   ALLOWED     | USED FOR  |         |
|          |             |   WRITES     |   FOR READS   |  READS    |         |
+----------+-------------+--------------+---------------+-----------+         |
|        5 |           0 |            0 |           2   |       0   |         |
+----------+-------------+--------------+---------------+-----------+---------+
+=============================================================================+
```

Figure 77. RLOG IMS DASD Logging Statistics report

Following are the fields in Part I, the Logging Environment:

OLDS DEFINED
The fixed number of OLDS defined during this interval.

OLDS STOPPED
The average number of OLDS stopped.

OLDS IN ERROR
The average number of OLDS found in error.

WADS DEFINED
The fixed number of WADS defined.

WADS USED
The average number of WADS used.

WADS SPARES
The average number of WADS, which are not used (WADS DEFINED - WADS USED).

OLDS LOGGING
This is either single or dual.

WADS LOGGING
: This is either single, dual, or inactive (inactive might result from WADS being smaller than OLDS buffer).

AUTO ARCHIVE LIMIT
: This is none (when not set) or the limit last seen for this interval.

Following are the fields in Part II, the Logging Statistics:

LOG RECORDS
: The cumulative number of log records written.

WRITE AHEAD RQSTS
: The cumulative number of write ahead requests.

DC WAITS FOR WRITE AHEAD
: The number of DC waits for write ahead. (This information is not applicable in a DBCTL environment.)

OUTPUT BUFFER WAITS
: The number of output buffers waits.

CHECKPOINT OUTPUT BUFFER WAITS
: The number of waits for a log buffer during checkpoint processing.

WADS EXCPVRS
: The cumulative number of WADS EXCPVRS.

 # 2K BLOCKS WADS
: The number of 2K blocks that are written for WADS.

OLDS BLOCKS WRITTEN
: The number of blocks that are written for OLDS.

OLDS BLOCKS READ
: The number of blocks that are read from OLDS.

 # SYSTEM CHECKPOINTS
: The cumulative number of system checkpoints.

Following are the fields in Part III, Olds Buffer Statistics:

DEFINED
: The fixed number of OLDS buffers defined.

AVAILABLE
: The average number of OLDS buffers available.

USED FOR WRITES
: The average number of OLDS buffers used for writes.

ALLOWED FOR READS
: The fixed number of OLDS buffers allowed for reads.

USED FOR READS
: The average number of OLDS buffers used for reads.

Displaying IMS Combined Fixed-length Pool Statistics (RCFP panel)

The IMS Combined Fixed Pool Statistics panel shows size and usage information for each IMS fixed buffer format pool.

The following example of an IMS Combined Fixed Pool Statistics report shows how the system displays IMS statistics that are based on the IMS fixed buffer format pool.

```
EPILOG/IMS       08/29/08  9:40  Mode: PAGE           1 of 3   LAST FRAME
CMD==>
*******************************************************************************
+=================== IMS Combined Fixed Pool Statistics =====================+
| Period:  09:18 to 09:29 on 08/29/08                  Elap =10:47 M     IMSD |
+-----------------------------------------------------------------------------+
|   CESS -- Communications External Subsys    SPAP -- Scratch Pad Area Pool   |
|   CIOP -- Communications I/O Pool           FPWP -- Fast Path Work Pool     |
|   HIOP -- Communications High I/O Pool      EMHB -- Expedited Msg Handler   |
|   LUMC -- LU6 Mgr Common Pool               LUMP -- LU6 Mgr Private Pool    |
|   AOIP -- Automated Operator Interface                                      |
+-----------------------------------------------------------------------------+
| Pool |Buffer |Primary|Second | Total | Free  |Blks since /CHE|Blocks |Blocks|
|      | Size  |Buf/Blk|Buf/Blk| Blocks| Blocks| High  |  Low  |Alloc  |Freed |
+------+-------+-------+-------+-------+-------+-------+-------+-------+------+
| CESS |   128 |    32 |    16 |     0 |     0 |     0 |     0 |     1 |    1 |
| CESS |   256 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS |   512 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS |  1024 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS |  2048 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS |  4096 |    16 |     8 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS |  8192 |     8 |     4 |     0 |     0 |     0 |     0 |     0 |    0 |
| CESS | 16384 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |    0 |
+------+-------+-------+-------+-------+-------+-------+-------+-------+------+
| SPAP |   264 |    64 |    32 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP |   520 |    64 |    32 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP |  1032 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP |  2056 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP |  4104 |    16 |     8 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP |  8200 |     8 |     4 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP | 16392 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |    0 |
| SPAP | 32776 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |    0 |
+------+-------+-------+-------+-------+-------+-------+-------+-------+------+
```

Figure 78. IMS Combined Fixed Pool Statistics report

The following are the fields in the RCFP report:

Buffer Size
Various buffer sizes as defined in the DFSSPMnn IMS.PROCLIB member

All values in the following columns refer to the blocks that contain this buffer size.

Primary Buf/Blk
Primary allocation of buffers as defined in the DFSSPMnn IMS.PROCLIB member

Secondary Buf/Blk
Secondary allocation of buffers as defined in the DFSSPMnn IMS.PROCLIB member

Total Blocks
Number of blocks currently in use or available

Free Blocks
Number of blocks currently available

High Blks since /CHE
High block count since the last IMS/CHECKPOINT command

Low Blks since /CHE
Low block count since the last IMS/CHECKPOINT command

Blocks Alloc
Number of blocks that are allocated as part of the IMS Buffer Pool Manager Expansion process

Blocks Freed
Number of blocks that are freed as part of the IMS Buffer Pool Manager Compression process

Automated Operator Interface (AOIP) pool

You can view a page of the RCFP report that shows statistics for the AOIP pool.

At the top of the first page of the RCFP panel (IMS Combined Fixed Pool Statistics report), note the addition of the AOIP pool. Figure 79 on page 114 shows a page of the RCFP report with statistics for the AOIP pool.

```
+================== IMS Combined Fixed Pool Statistics ================
| EMHB |   520 |    64 |    32 |     0 |     0 |     0 |     0 |     0 |
| EMHB |  1032 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| EMHB |  2056 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| EMHB |  4104 |    16 |     8 |     0 |     0 |     0 |     0 |     0 |
| EMHB |  8200 |     8 |     4 |     0 |     0 |     0 |     0 |     0 |
| EMHB | 16392 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |
| EMHB | 32776 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |
+------+-------+-------+-------+-------+-------+-------+-------+-------+
| LUMC |   136 |    32 |    32 |     0 |     0 |     0 |     0 |     0 |
| LUMC |   264 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMC |   520 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMC |  1032 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMC |  2056 |    16 |     8 |     0 |     0 |     1 |     0 |     1 |
| LUMC |  3072 |    12 |    12 |     0 |     0 |     0 |     0 |     0 |
| LUMC |  4104 |     8 |     8 |     0 |     0 |     0 |     0 |     0 |
| LUMC | 33032 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |
+------+-------+-------+-------+-------+-------+-------+-------+-------+
| LUMP |   136 |    32 |    32 |     0 |     0 |     0 |     0 |     0 |
| LUMP |   264 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMP |   520 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMP |  1032 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| LUMP |  2056 |    16 |     8 |     1 |     0 |     1 |     0 |     1 |
| LUMP |  3072 |    12 |    12 |     1 |     0 |     1 |     0 |     1 |
| LUMP |  4104 |     8 |     8 |     0 |     0 |     0 |     0 |     0 |
| LUMP | 33032 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |
+------+-------+-------+-------+-------+-------+-------+-------+-------+
| AOIP |   136 |    32 |    32 |     0 |     0 |     0 |     0 |     0 |
| AOIP |   264 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| AOIP |   520 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| AOIP |  1032 |    32 |    16 |     0 |     0 |     0 |     0 |     0 |
| AOIP |  2056 |    16 |     8 |     1 |     0 |     1 |     0 |     1 |
| AOIP |  3072 |    12 |    12 |     1 |     0 |     1 |     0 |     1 |
| AOIP |  4104 |     8 |     8 |     0 |     0 |     0 |     0 |     0 |
| AOIP | 33032 |     4 |     2 |     0 |     0 |     0 |     0 |     0 |
+=====================================================================+
```

Figure 79. IMS Combined Fixed Pool Statistics report (page 3)

Displaying IMS Combined Pool Statistics (RCMP panel)

The IMS Combined Pool Statistics panel shows size and usage information for the IMS pools that are listed in its header. The MAIN pool work area size is critical when a varying number of main storage conversations can be in process.

The following example shows how the system displays IMS combined pool statistics only for variable-length buffer pools.

```
EPILOG/IMS       08/29/08   9:45   Mode: PAGE
CMD==>
*********************************************************************************
+===================== IMS Combined Pool Statistics ==========================+
| Period:  09:18 to 09:29 on 08/29/08                  Elap =10:47 M      IMSD |
+------------------------------------------------------------------------------+
|   MAIN -- General Work Area Pool (WKAP)     DMBW -- SMB Work Pool            |
|   EPCB -- Extended PCB Pool                 PSBW -- PSB Work Pool            |
+------------------------------------------------------------------------------+
| Pool |    Total    |    Free    |   Usage   |   Number  | Largest  | Utilization|
|      |    Size     |    Space   | Highwater |  Free Blk | Free Blk |    Percent |
+------+-------------+------------+-----------+-----------+----------+------------+
|MAIN  |      49152 |      49152 |       0    |      1    |    49152 |        .0  |
+------+-----------+-----------+-----------+-----------+-----------+-------------+
|DMBW  |      24576 |      24576 |       0    |      1    |    24576 |        .0  |
+------+-----------+-----------+-----------+-----------+-----------+-------------+
|PSBW  |      24576 |        208 |    3496    |      1    |      208 |      99.2  |
+------+-----------+-----------+-----------+-----------+-----------+-------------+
|EPCB  |      12288 |      10280 |    1520    |      1    |    10280 |      16.3  |
+------+-----------+-----------+-----------+-----------+-----------+-------------+
+=============================================================================+
```

Figure 80. Display RCMP—Combined Pool Statistics

The following are the fields in the RCMP panel.

Total Size
The fixed maximum space that is allowed for the pool for this interval.

Free Space
The average amount of the total space that was free for this period.

Usage Highwater
The maximum amount of used space that is seen for this interval.

Number Free Blks
The average number of free blocks that are seen for this period.

Largest Free Blk
The maximum number of free blocks that are seen for this period.

Utilization Percent
The percentage of pool space that is used during this interval.

Displaying IMS CIOP Statistics (RCIP panel)

The IMS CIOP Statistics panel provides information about the communications I/O pool.

The following example of an IMS CIOP report shows how the system displays IMS statistics that are based on the size and usage information for each IMS fixed buffer format pool.

```
+======================== IMS CIOP Statistics ===========================+
|Period:  11:45 to 12:00 on 08/29/08              Elap =15:00 M     IMSD |
+------------------------------------------------------------------------+
|                   CIOP -- Communications I/O Pool                      |
+------------------------------------------------------------------------+
|       | Total  | Alloc  |  Free  |  High  | RecAny | Linebuf | Poolavl |
+-------+--------+--------+--------+--------+--------+---------+---------+
|Storage| 272450 | 101114 | 171336 |  47800 |  33280 |   129   |  67735  |
+-------+--------+--------+--------+--------+--------+---------+---------+
|Percent|   na   |  16.3% |  83.6% |  23.3% |  16.2% |    0%   |    0%   |
+-------+--------+--------+--------+--------+--------+---------+---------+
|Blocks |   na   |   10   |    1   |   na   |    8   |    1    |    0    |
+-------+--------+--------+--------+--------+--------+---------+---------+
+========================================================================+
```

Figure 81. IMS CIOP Statistics report

The RCIP panel contains the following fields:

Total
The total amount of space available for the CIOP pool for this interval (ALLOC + FREE).

Alloc
The total amount of space that is allocated for CIOP buffer use (RECANY + LINEBUF + POOLAVL).

Free
The total amount of space available for use.

High
The highest amount of used space that is seen for this interval.

RecAny
The total amount of space that is used for VTAM terminal input.

Linebuf
The total amount of space that is used for BTAM terminal I/O and VTAM terminal output.

Poolavl
The amount of space available after the allocated space.

Displaying IMS DMB Pool Statistics (RDMP panel)

The IMS DMB Pool Statistics panel provides size and usage information about the Data Management Block pool.

The following example of an RDMP IMS DMB Pool report shows how the system displays IMS statistics that are based on the DMB (Data Management Block) Pool.

```
+========================== IMS DMB Pool Statistics ==========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|                        DMB (Data Management Block) Pool                     |
+-----------------------------------------------------------------------------+
|       | Total   | Alloc   | Active |  Free   |  High   |Resident|Res-Act|G-Total|
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Storage| 112640 |  89088 |  14336 |  16384 | 107520 |   7800 |   0   | 120440|
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Percent|   na   |  79.1% |  12.7% |  20.9% |  95.4% |    na  |   na  |   na  |
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Blocks |   110  |    0   |    0   |    1   |   na   |    9   |   na  |   na  |
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|         Current Utilization: 87.5%        Highest Utilization: 98.4%        |
+=============================================================================+
```

Figure 82. RDMP IMS DMB Pool report

The following are the fields in the RDMP panel.

Total
Total space in pool for non-resident DMBs. This fixed amount does not change during the interval.

Alloc
The average of the allocated space in pool.

Active
The average of the active allocated space in pool.

Free
The average of the deallocated space in pool.

High
The maximum space that is ever allocated in pool.

Resident
The total space outside pool for resident DMBs.

Res-Act
The average of the active space for resident DMBs.

G-Total
This is the sum of the total and resident DMB pools.

Displaying IMS PSB Pool Statistics (RPSP panel)

The IMS PSB Pool Statistics panel provides size and usage information about the Program Specification Block pool.

The following example of an IMS PSB Pool Statistics report shows how the system displays IMS statistics that are based on the Program Specification Block pool.

```
+========================= IMS PSB Pool Statistics =========================+
| Period:  11:45 to 12:00 on 08/29/08                 Elap =15:00 M     IMSD |
+---------------------------------------------------------------------------+
|                    PSB (Program Specification Block) Pool                 |
+---------------------------------------------------------------------------+
|       | Total  | Alloc  | Active |  Free  |  High  |Resident|Res-Act|G-Total|
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Storage|  53248 |  11232 |      0 |  42016 |  53240 |      0 |     0 |  53248|
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Percent|   na   |  21.1% |    .0% |  78.9% | 100.0% |    na  |   na  |   na  |
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|Blocks |   na   |     21 |      0 |      1 |      0 |    na  |   na  |   na  |
+-------+--------+--------+--------+--------+--------+--------+-------+-------+
|         Current Utilization:    .0%       Highest Utilization: 100.0%       |
+---------------------------------------------------------------------------+
+---------------------------------------------------------------------------+
|                 DLISAS PSB (Program Specification Block) Pool             |
+---------------------------------------------------------------------------+
|       | Total  | Alloc  | Active |  Free  |  High  |Resident|Res-Act|G-Total|
+-------+--------+--------+--------+--------+--------+--------+-------+-------|
|Storage|  81920 |  81920 |  81920 |      0 |  81920 |      0 |     0 |  81920|
+-------+--------+--------+--------+--------+--------+--------+-------+-------|
|Percent|   na   | 100.0% | 100.0% |   0.0% | 100.0% |    na  |   na  |   na  |
+-------+--------+--------+--------+--------+--------+--------+-------+-------|
|Blocks |   na   |     40 |     40 |      0 |     40 |    na  |   na  |   na  |
+-------+--------+--------+--------+--------+--------+-------------------------|
|         Current Utilization: 100.0%       Highest Utilization: 100.0%       |
+---------------------------------------------------------------------------+
+===========================================================================+
```

Figure 83. IMS PSB Pool Statistics report

The following are the fields in the RPSP panel:

Total
: The amount of fixed space that might be allocated for this pool during this interval (non-resident PSBs only).

Alloc
: The average amount of space that is allocated for this pool.

Active
: The amount of allocated space that is active.

Free
: The amount of allocated space that is free.

High
: The maximum amount of space to be active.

Resident
: The amount of space that is allocated outside the PSB pool for resident PSBs.

Res-Act
: The amount of space that is allocated to the resident PSBs.

G-Total
: This is the total + resident.

Current Utilization
: This is the same as active percent.

Highest Utilization
: This is the same as high percent.

Displaying IMS QBUF Statistics (RQBP panel)

The IMS QBUF Statistics panel provides size and usage information about the Message Queue Buffer pool.

The following example of an IMS QBUF Statistics report shows how the system displays IMS statistics that are based on the Message Queue Buffer pool.

```
+=========================== IMS QBUF Statistics ============================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+---------------------------------------------------------------------------+
|                       QBUF -- Message Queue Buffer Pool                    |
+---------------------------------------------------------------------------+
|           |   Total     |  Short   |  Long    |  Qblks  |   Not    |  Block  |
|           |             |  Message |  Message |         |  Active  |  Size   |
+-----------+-------------+----------+----------+---------+----------+---------+
|  Storage  |     40320   |   12096  |    8064  |   4032  |    16128 |   4032  |
+-----------+-------------+----------+----------+---------+----------+---------+
|  Percents |        na   |   30.0%  |   20.0%  |  10.0%  |    40.0% |    na   |
+-----------+-------------+----------+----------+---------+----------+---------+
|  Blocks   |        10   |       3  |       2  |      1  |        4 |    na   |
+-----------+-------------+----------+----------+---------+----------+---------+
+---------------------------------------------------------------------------+
|                                 QBUF Requests                              |
+---------------------------------------------------------------------------+
|           |   Total     |  Found   | Enqueue/ | Enqueue | Dequeue  |  IWAITS |
|           |  Requests   | in pool  | Dequeue  |         |          |         |
+-----------+-------------+----------+----------+---------+----------+---------+
|  Counts   |      6928   |   6121   |     630  |    418  |     212  |       0 |
+-----------+-------------+----------+----------+---------+----------+---------+
+---------------------------------------------------------------------------+
|                                   QBUF I/Os                                |
+---------------------------------------------------------------------------+
|           |     Total     |    Reads   |   Writes   |    Forced  |   Chkpt    |
|           |     I/Os      |            |            |    Writes  |   Writes   |
+-----------+---------------+------------+------------+------------+------------+
|  Counts   |       197     |      124   |       73   |      13    |      60    |
+-----------+---------------+------------+------------+------------+------------+
+============================================================================+
```

Figure 84. IMS QBUF Statistics report

The RQBP panel contains the following fields:

Total
The total space that is assigned for the message queue buffer pool (Short Message + Long Message + QBLKS + Not Active).

Short Message
The space that is allocated for the short message blocks.

Long Message
The space that is allocated for the long message blocks.

Qblks
The space that is allocated for blocks for the QBLKS data set.

Not Active
The space that is not allocated for any of the preceding categories.

Block Size
The fixed size for all the QBUF blocks.

Total Requests
The total requests issued to the queue manager.

Found in pool
The requests that are satisfied internally without having to complete any I/O.

Enqueue/Dequeue
The sum of enqueued and dequeued messages.

Enqueue
The number of enqueued messages.

Dequeue
The number of dequeued messages (ENQ + DEQ).

IWAITS
The number of IWAITS issued when buffers are unavailable.

Total I/Os
The total I/Os issued for read and write requests.

Reads
The number of I/Os issued for all read requests.

Writes
The number of I/Os issued for all write requests.

Forced Writes
The number of forced writes.

Chkpt Writes
The number of checkpoint writes (writes - forced writes).

Displaying IMS Message Format Buffer Pool Statistics (RMFP panel)

The IMS Message Format Buffer Pool Statistics panel provides size and activity information about the Message Format Buffer pool.

The following example of an IMS Message Format Buffer Pool Statistics report shows how the system displays IMS statistics that are based on the Message Format Buffer pool.

```
+================= IMS Message Format Buffer Pool Statistics =================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|                  Message Format Buffer Pool Storage Part I                  |
+-----------------------------------------------------------------------------+
| Fixed | Dynamic | FRE Blks | Dir Index | Dir Indx | Tot Dyn | Dyn | Dyn Dir |
|       |         |          |   Size    | Entries  | Dir Size| Dir | Entries |
+---------+---------+----------+-----------+----------+---------+---+----------+
|  1072 |  31548  |      200 |       390 |      325 |    70154 | 10 |     178 |
+---------+---------+----------+-----------+----------+---------+---+----------+
+-----------------------------------------------------------------------------+
|                  Message Format Buffer Pool Storage Part II                 |
+-----------------------------------------------------------------------------+
|  Type    |  Total   | Loading   | Immediate | Prefetch |   Free   | Inactive  |
|          |          |           |   Fetch   |          |          |           |
+---------+---------+----------+-----------+----------+---------+------------+
| Storage |  120832 |   24576   |   22528   |   16384  |   20480  |   10240   |
+---------+---------+----------+-----------+----------+---------+------------+
| Percents|    na    |   20.3%   |   18.6%   |   13.6%  |   16.9%  |    8.5%   |
+---------+---------+----------+-----------+----------+---------+------------+
| Blocks   |      59 |       12  |        11 |        8 |       10 |        5  |
+---------+---------+----------+-----------+----------+---------+------------+
+-----------------------------------------------------------------------------+
|                     Message Format Buffer Pool Requests                     |
+-----------------------------------------------------------------------------+
|  Total   |Immediate| Prefetch | Found in  |Immediate |  Blocks  | Prefetch  |
|          |  Fetch  |          |   Pool    |Fetch Wait|  Washed  | Ignored   |
+---------+---------+----------+-----------+----------+---------+------------+
|  290825 |  243187 |    47638 |     23487 |        0 |     2039 |        0  |
+---------+---------+----------+-----------+----------+---------+------------+
+-----------------------------------------------------------------------------+
|                       Message Format Buffer Pool I/Os                       |
+-----------------------------------------------------------------------------+
|  Total   |Immediate| Prefetch | Directory |   Not    |   I/O    |           |
|          |  Fetch  |          |           |  Found   |  Errors  |           |
+---------+---------+----------+-----------+----------+---------+------------+
|   59643 |   54267 |     5376 |       231 |      176 |        0 |           |
+---------+---------+----------+-----------+----------+---------+------------+
+=============================================================================+
```

Figure 85. IMS Message Format Buffer Pool Statistics report

Following are the fields that are found in the Message Format Buffer Pool Storage Part I:

Fixed
The amount of fixed space available for non-MFS definitions for this interval.

Dynamic
The dynamic amount of space available for MFS definitions.

FRE blks
: The fixed amount of space available for fetch request elements, which anchor each block in the pool (the pool is partially filled only if there are not enough FREs).

Dir Index Size
: The fixed size of the directory index. The directory is searched after the FREs and $$IMSDIR are searched.

Dir Index Entries
: The fixed number of entries in the directory index.

Tot Dyn Dir Size
: The total amount of space that is used for the dynamic directories.

Dyn Dir
: An incore area IMS dynamically creates to manage MFS formats.

Dyn Dir Entries
: The number of format blocks that are represented by the current dynamic directory.

Following are the fields that are found in the Message Format Buffer Pool Storage Part II:

Type
: The type of data in the column.

Total
: The maximum amount of space for this pool.

Loading
: The average amount of space that is used for prefetch while loading.

Immediate Fetch
: The average amount of space that is used for immediate fetch while loading.

Prefetch
: The cumulative number of prefetch I/Os.

Free
: The average amount of space available

Inactive
: The average amount of space that is inactive because MFS blocks are in-core, that is, are no longer used.

Following are the fields that are found in the Message Format Buffer Pool Requests:

Total
: The cumulative number of total block requests.

Immed Fetch
: The cumulative number of immediate fetch block requests.

Prefetch
: The cumulative number of prefetch block requests.

Found in Pool
: The cumulative number of blocks that are found within pool.

Immed Fetch Wait
: The cumulative number of immediate fetch waits before prefetch completed.

Blocks Washed
: The cumulative number of times a block is freed (this might be the result of a possible FRE shortage).

Prefetch Ignored
: The cumulative number of times a prefetch is ignored (this might be the result of a possible FRE shortage).

Following are the fields in the Message Format Buffer Pool I/O:

Total
: The maximum space for this pool.

Immed Fetch
The cumulative number of immediate fetch I/Os.

Prefetch
The cumulative number of prefetch I/Os.

Directory
The cumulative number of index directory I/Os.

Not Found
The cumulative number of I/Os for not found entries.

I/O Errors
The cumulative number of I/O errors.

Displaying IMS PI Enqueue Pool Statistics (RPIP panel)

The IMS PI Enqueue Pool Statistics panel provides size and usage information about the PI enqueue pool.

The following example of an IMS PI Enqueue Pool Statistics report shows how the system displays IMS statistics that are based on the PI enqueue pool.

```
+====================== IMS PI Enqueue Pool Statistics ======================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+---------------------------------------------------------------------------+
|                     PINQ  (Program Isolation Enqueue Pool)                  |
+---------------------------------------------------------------------------+
|   Current  |   Maximum   |     Pool      |   Free Space   |   Utilization   |
|     Size   |     Size    |   Increment   |    in bytes    |     Percent     |
+------------+-------------+---------------+----------------+-----------------+
|    53744   |    409600   |     4096      |     55261      |       3.6%      |
+------------+-------------+---------------+----------------+-----------------+
+===========================================================================+
```

Figure 86. IMS PI Enqueue Pool Statistics report

The RPIP panel contains the following fields:

Current Size
The average size of the PI enqueue pool during the interval (this value starts at zero).

Maximum Size
The maximum size the pool can increase to.

Pool Increment
The amount that the current size increments by until the maximum size is reached.

Free Space in bytes
The average amount of space within the current size available during interval.

Utilization Percent
The percent of the current size that is used during interval; this interval is calculated as follows (current size - free space)/current size.

Displaying IMS SAP Pool Statistics (RSAP panel)

The IMS SAP Pool Statistics panel provides usage information about the Save Area Prefix pool.

The following example of an RSAP IMS SAP Pool Statistics report shows how the system displays IMS statistics that are based on the Save Area Prefix pool.

```
+========================== IMS SAP Pool Statistics ==========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|                          SAP (Save Area Prefix) Pool                        |
+-----------------------------------------------------------------------------+
| Dynamic SAPs allocated =  22  Dynamic SAPs in use =  1  Utilization =  4.5% |
+=============================================================================+
```

Figure 87. RSAP IMS SAP Pool Statistics report

The following are the fields in the RSAP panel:

Dynamic SAPs allocated
The average number of dynamic SAPs allocated during the interval.

Dynamic SAPs in use
The number of allocated SAPs that are used during the interval.

Utilization
The percentage of dynamic SAPs that are used during the interval.

Displaying IMS FRE Pool Statistics (RFRP panel)

The IMS FRE Pool Statistics panel provides usage information about the Fetch Request Element pool.

The following example of an RFRP IMS FRE Pool Statistics report shows how the system displays IMS statistics that are based on the Fetch Request Element pool.

```
+========================= IMS FRE Pool Statistics ===========================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|                       FRE (Fetch Request Element) Pool                      |
+-----------------------------------------------------------------------------+
|  Fixed FREs allocated =  40   Fixed FREs in use =  10   Utilization = 25.0% |
+-----------------------------------------------------------------------------+
|                                Dynamic FREs =  0                            |
+=============================================================================+
```

Figure 88. RFRP IMS FRE Pool Statistics report

The following are the fields in the RFRP panel:

Fixed FREs allocated
The fixed number of FREs that were allocated when the message format block pool is built by using the IMS “FRE=” startup parameter.

Fixed FREs in use
The average number of allocated fixed FREs that are used during the interval.

Utilization
The percentage of fixed FREs that are used during the interval.

Dynamic FREs
The average number of FREs reserved from MFS buffer pool when fixed FREs are 100% used.

Displaying IMS Virtual Storage Option Statistics (RVSO panel)

The Fastpath VSO Dataspace Statistics panel provides usage information about the VSO areas.

The following example of a fast path VSO Dataspace Statistics report shows how the system displays IMS virtual storage option statistics that are based on the fast path VSO dataspace.

```
+--------------------- Fastpath VSO Dataspace Statistics----------------------+
| Period:  08:50 to 08:59 on 08/29/08                  Elap = 8:19 M     I51N |
+-----------------------------------------------------------------------------+
| Dataspace Name: 00000FPV                                                    |
+-----------------------------------------------------------------------------+
|  VSO Area |   DSP    |   DSP    |   DSP    |   DASD   |   DASD   |   DASD   |
|    Name   |  Reads   |  Writes  |   Rate   |  Reads   |  Writes  |   Rate   |
|-----------+----------+----------+----------+----------+----------+----------+
|  DFSIVD3B |      100 |      150 |     0.20 |       30 |       60 |     0.10 |
|  DFSIVD3D |      120 |      175 |     0.33 |       41 |       72 |     0.10 |
+-----------------------------------------------------------------------------+
| Dataspace Name: 00001FPV                                                    |
+-----------------------------------------------------------------------------+
|  VSO Area |   DSP    |   DSP    |   DSP    |   DASD   |   DASD   |   DASD   |
|    Name   |  Reads   |  Writes  |   Rate   |  Reads   |  Writes  |   Rate   |
|-----------+----------+----------+----------+----------+----------+----------+
|  DFSIVD4A |      103 |      127 |     0.26 |       31 |       43 |     0.08 |
|  DFSIVD4B |      210 |      301 |     0.57 |       11 |       21 |     0.04 |
|  DFSIVD4C |      101 |      151 |     0.28 |       33 |       63 |     0.11 |
+-----------------------------------------------------------------------------+
|                         Dataspace Summary -- All Areas                      |
+-----------------------------------------------------------------------------+
| Number of Dataspaces |        2 | Number of Areas     |        5 |          |
| High DSP  I/O Name   | 00001FPV | High DSP  I/O Rate  |     0.57 |          |
| High Area I/O Name   | DFSIVD4B | High Area I/O Rate  |     0.57 |          |
| Low DSP   I/O Name   | 00000FPV | Low DSP   I/O Rate  |     0.20 |          |
| Low Area  I/O Name   | DFSIVD3B | Low Area  I/O Rate  |     0.20 |          |
+-----------------------------------------------------------------------------+
+=============================================================================+
```

Figure 89. Fast path VSO Dataspace Statistics report

The following are the fields in the RVSO panel:

VSO Area Name
Name of the fast path Virtual Storage Option area for which statistics are collected.

DSP Reads
Number of dataspace reads done on behalf of the VSO area.

DSP Writes
Number of dataspace writes done on behalf of the VSO area.

DSP Rate
Total DSP Reads plus DSP Writes calculated into a rate per second.

DASD Reads
Total reads from DASD done on behalf of the VSO area.

DASD Writes
Total writes to DASD on behalf of the VSO area.

DASD Rate
Total DASD reads plus DASD writes calculated into a rate per second.

Displaying IMS VSAM Buffer Pool Statistics (RVSP panel)

The IMS VSAM Buffer Pool Statistics panel provides size and usage information about the IMS VSAM buffer pools, which are classified by subpools.

The following example of the IMS VSAM Buffer Pool Statistics report shows how the system displays IMSstatistics based on the IMS VSAM buffer pools.

```
+==================== IMS VSAM Buffer Pool Statistics ======================+
| Period:  19:30 to 19:45 on 08/29/08                  Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
|            VSAM Subpool # 1 - 10 buffers of 1024 bytes each                |
+-----------------------------+---------+-------------------------+-----------+
|       Pool ID               |    POL1 |   Pool number           |     0     |
|       Subpool type          |    Data |   Subpool # within pool |     1     |
|       Buffers Page-Fixed?   |      No |   IOBs Page-Fixed?      |    No     |
|       Retrieves by RBA      |   23876 |   Retrieves by Key      |   123     |
|       Records altered       |    8763 |   Records created       |   109     |
|       VSAM Reads            |    1298 |   VSAM Writes           |    63     |
|       Found in pool         |    1098 |   Sync points taken     |     1     |
|       Buffers in error      |       0 |   Maximum errors        |     0     |
|       Hit ratio             |   84.5% |   Backed by hiperspace? |    No     |
+-----------------------------+---------+-------------------------+-----------+
+----------------------------------------------------------------------------+
|            VSAM Subpool # 2 -  9 buffers of 2048 bytes each                |
+-----------------------------+---------+-------------------------+-----------+
|       Pool ID               |    POL1 |   Pool number           |     0     |
|       Subpool type          |    Data |   Subpool # within pool |     2     |
|       Buffers Page-Fixed?   |      No |   IOBs Page-Fixed?      |    No     |
|       Retrieves by RBA      |    7654 |   Retrieves by Key      |  2671     |
|       Records altered       |    1987 |   Records created       |   732     |
|       VSAM Reads            |     298 |   VSAM Writes           |   287     |
|       Found in pool         |     109 |   Sync points taken     |    16     |
|       Buffers in error      |       0 |   Maximum errors        |     0     |
|       Hit ratio             |   36.5% |   Backed by hiperspace? |    No     |
+-----------------------------+---------+-------------------------+-----------+
+----------------------------------------------------------------------------+
|            VSAM Subpool # 3 - 20 buffers of 4096 bytes each                |
+-----------------------------+---------+-------------------------+-----------+
|       Pool ID               |    POL1 |   Pool number           |     0     |
|       Subpool type          |    Data |   Subpool # within pool |     3     |
|       Buffer size           |    4096 |   Number of buffers     |    20     |
|       Buffers Page-Fixed?   |      No |   IOBs Page-Fixed?      |    No     |
|       Retrieves by RBA      |    7654 |   Retrieves by Key      |  2671     |
|       Records altered       |    1987 |   Records created       |   732     |
|       VSAM Reads            |     298 |   VSAM Writes           |   287     |
|       Found in pool         |     109 |   Sync points taken     |    16     |
|       Buffers in error      |       0 |   Maximum errors        |     0     |
|       Hit ratio             |   53.7% |   Backed by hiperspace? |   Yes     |
|     Databases assigned to subpool:                                         |
|       XDCPA123 XDCPA125 XDCPA985                                           |
+-----------------------------+---------+-------------------------+-----------+
+----------------------------------------------------------------------------+
|            VSAM Subpool # 1 - 20 buffers of 4096 bytes each                |
+-----------------------------+---------+-------------------------+-----------+
|       Pool ID               |    POL2 |   Pool number           |     1     |
|       Subpool type          |   Index |   Subpool # within pool |     1     |
|       Buffers Page-Fixed?   |      No |   IOBs Page-Fixed?      |    No     |
|       Retrieves by RBA      |    7654 |   Retrieves by Key      |  2671     |
|       Records altered       |    1987 |   Records created       |   732     |
|       VSAM Reads            |     298 |   VSAM Writes           |   287     |
|       Found in pool         |     109 |   Sync points taken     |    16     |
|       Buffers in error      |       0 |   Maximum errors        |     0     |
|       Hit ratio             |   36.5% |   Backed by hiperspace? |    No     |
+-----------------------------+---------+-------------------------+-----------+
|                   VSAM Pool Summary -- All Subpools                        |
+-----------------------------+---------+-------------------------+-----------+
|  Retrieves by RBA           |   57950 |   Retrieves by Key      |  9404     |
|  Records altered            |   14637 |   Records created       |  5648     |
|  VSAM Reads                 |    3772 |   VSAM Writes           |  1442     |
|  Found in pool              |    2325 |   Sync points taken     |    64     |
|  Buffers in error           |       0 |   Maximum errors        |     0     |
|  Hit ratio                  |   61.6% |                         |           |
+-----------------------------+---------+-------------------------+-----------+
+============================================================================+
```

Figure 90. IMS VSAM Buffer Pool Statistics report

The following are the fields in the RVSP panel:

VSAM Subpool #
: The internally assigned subpool number. Subpools are numbered sequentially beginning with one. In addition to the subpool number, this line also provides the number and size of the buffers that make up the subpool.

Pool ID
Pool name that is assigned by the DFNVSMxx member.

Pool number
The pool number that is assigned internally by IMS.

Subpool type
Indicates whether the subpool is an index or data subpool, or whether this field is not applicable.

Subpool # within pool
Number of the subpool within the pool.

Buffers Page-Fixed?
Indicates whether the buffers are page fixed for the interval.

IOBs Page-Fixed?
Indicates whether the VSAM I/O buffers are page-fixed for the interval.

Retrieves by RBA
The cumulative number of retrieves that are issued by RBA.

Retrieves by Key
The cumulative number of retrieves that are issued by key.

Records altered
The cumulative number of records that are altered.

Records created
The cumulative number of ESDS inserts and KSDS inserts.

VSAM Reads
The cumulative number of reads that are completed by VSAM.

VSAM writes
The cumulative number of user initiated writes and non-user initiated writes.

Found in pool
The cumulative number of buffers that are found in pool.

Sync points taken
The cumulative number of sync points taken.

Buffers in error
The cumulative number of error buffers found.

Maximum errors
The number of errors that are generated during writes and reads for the subpools and the total errors for the combined subpools.

Hit ratio
The ratio of records that are found in the pool to VSAM reads.

Backed by Hiperspace?
Reports whether the buffer subpool is backed by Hiperspace™.

Databases assigned to this subpool
EPILOG reports on multiple VSAM database subpool, displaying the databases that are allocated to each specific VSAM subpool.

To limit the scope of the RVSP panel, add a suboperand. Examples are shown as follows:

DISPLAY RVSP (1 3)
Limits scope of display to VSAM subpools 1 and 3.

DISPLAY RVSP (2:6)
Limits scope of display to VSAM subpools 2 through 6; the colon (:) denotes *range*.

Displaying IMS ISAM/OSAM Buffer Pool Statistics (ROSP panel)

The IMS ISAM/OSAM Buffer Pool Statistics panel provides size and usage information about the IMS ISAM/OSAM buffer pools, which are classified by subpools.

The following example of an IMS ISAM/OSAM Buffer Pool Statistics report shows how the system displays IMS statistics that are based on the IMS ISAM/OSAM buffer pools.

```
+==================== IMS ISAM/OSAM Buffer Pool Statistics ===================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
| ISAM/OSAM Subpool # 1    Subpool ID: None    4 buffers of 1024 bytes each   |
+--------------------------+------------+--------------------------+------------+
|   Buffers Page-fixed?    |         No |   Prefix Page-fixed?     |         No |
|   Locate calls           |      23876 |   Locates found in pool  |       1235 |
|   Read requests          |       8763 |   Single block writes    |       1092 |
|   New Block requests     |       1298 |   Records altered        |        634 |
|   Purge requests         |       1098 |   Blks written by purge  |         16 |
|   Total I/O errors       |          0 |   Buffers locked by errs |          0 |
|   Hit ratio              |       5.1% |                          |            |
+--------------------------+------------+--------------------------+------------+
+-----------------------------------------------------------------------------+
| ISAM/OSAM Subpool # 2    Subpool ID: None    4 buffers of 2048 bytes each   |
+--------------------------+------------+--------------------------+------------+
|   Buffers Page-fixed?    |         No |   Prefix Page-fixed?     |         No |
|   Locate calls           |      27654 |   Locates found in pool  |       2671 |
|   Read requests          |       1987 |   Single block writes    |       1732 |
|   New Block requests     |       1298 |   Records altered        |        287 |
|   Purge requests         |       1109 |   Blks written by purge  |         16 |
|   Total I/O errors       |          0 |   Buffers locked by errs |          0 |
|   Hit ratio              |       9.6% |                          |            |
+--------------------------+------------+--------------------------+------------+
+-----------------------------------------------------------------------------+
| ISAM/OSAM Subpool # 3    Subpool ID: AYCD    4 buffers of 2048 bytes each   |
+--------------------------+------------+--------------------------+------------+
|   Buffers Page-fixed?    |         No |   Prefix Page-fixed?     |         No |
|   Locate calls           |      18766 |   Locates found in pool  |       2827 |
|   Read requests          |       2398 |   Single block writes    |       3092 |
|   New Block requests     |       1878 |   Records altered        |        234 |
|   Purge requests         |       1009 |   Blks written by purge  |         16 |
|   Total I/O errors       |          0 |   Buffers locked by errs |          0 |
|   Hit ratio              |      15.0% |                          |            |
+--------------------------+------------+--------------------------+------------+
| Databases assigned to this subpool:                                          |
+----------+----------+---+------+-----+----+----------+----------+-----------+
| AYCD01FX | DB$HFS01 | DB$MDL03 | SDBHINDX |          |          |           |
+----------+----------+---+------+-----+----+----------+----------+-----------+
+-----------------------------------------------------------------------------+
|                 ISAM/OSAM Subpool Summary -- All Subpools                    |
+--------------------------+------------+--------------------------+------------+
|   Locate calls           |     127638 |   Locates found in pool  |     119827 |
|   Read requests          |      28376 |   Single block writes    |      27687 |
|   New Block requests     |      23817 |   Records altered        |       1872 |
|   Purge requests         |      18278 |   Blks written by purge  |         16 |
|   Total I/O errors       |          0 |   Buffers locked by errs |          0 |
|   Hit ratio              |      93.8% |                          |            |
+--------------------------+------------+--------------------------+------------+
+=============================================================================+
```

Figure 91. IMS ISAM/OSAM Buffer Pool Statistics report

The ROSP panel contains the following fields:

Buffers Page-Fixed?
: Indicates whether the buffers are page fixed for the interval.

Prefix Page-Fixed?
: Indicates whether the buffer prefix is page fixed for the interval.

Locate calls
: The cumulative number of locate calls.

Locates found in pool
: The cumulative number of locates found in pool.

Read requests
The cumulative number of read requests.

Single block writes
The cumulative number of single block writes.

New Block requests
The cumulative number of new block requests.

Records altered
The cumulative number of logical records altered.

Purge requests
The cumulative number of purge requests.

Blks written by purge
The cumulative number of blocks that are written per purge operation.

Total I/O errors
The cumulative number of I/O errors.

Buffers locked by errors
The cumulative number of locked buffers because of I/O

Hit ratio
The ratio of locate calls to locates found in the pool.

Subpool ID
An OSAM database can be assigned to a generic or specific subpool

Databases assigned to this subpool
EPILOG reports on multiple OSAM database subpools, displaying the databases that are allocated to each specific OSAM subpool.

To limit the scope of the ROSP panel, add a suboperand. For example:

DISPLAY ROSP (1 3)
Limits scope of display to ISAM/OSAM subpools 1 and 3.

DISPLAY ROSP (2:6)
Limits scope of display to ISAM/OSAM subpools 2 through 6; the colon (:) denotes *range*.

Displaying Database Error Statistics (RDBE panel)

The Database Error Statistics panel shows information about database I/O errors and database dynamic backout errors.

The following example of the Database Error Statistics report shows how the system displays IMSstatistics based on the database errors.

```
+======================== DATABASE ERROR STATISTICS ==========================+
| PERIOD:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
| PART 1 - DATABASE I/O ERRORS                                                |
+-----------------------------------------------------------------------------+
|   Database      |   DD Name      |    Type of Error   |       RBA           |
+-----------------+----------------+--------------------+---------------------+
|   DBXPARTS      |   DB000015     |     WRITE          |     12547360        |
+-----------------+----------------+--------------------+---------------------+
|   DBXINV02      |   DB000017     |     READ           |     10778000        |
+-----------------+----------------+--------------------+---------------------+
|   DBXINV03      |   DB000018     |     DBRC           |     10779000        |
+-----------------+----------------+--------------------+---------------------+
+-----------------------------------------------------------------------------+
| PART 2 - DATABASE DYNAMIC BACKOUT ERRORS                                    |
+-----------------------------------------------------------------------------+
|      Database        |       PSB Name     |      Date       |      Time      |
+----------------------+--------------------+-----------------+----------------+
|      DBXINV05        |       RMT00005     |      86010      |    13:00:32    |
+----------------------+--------------------+-----------------+----------------+
+=============================================================================+
```

Figure 92. Database Error Statistics report

Following are the fields in the Part I, Database I/O Errors:

Database
Name of the database

DD Name
DL/I ddname or fast path DEDB area name

Type of Error
Type of error that is associated with the database record is one of the following errors:

HOT I/O
Error for XRF standby system

READ
Read error

WRITE
Write error

DBRC
Record that is deactivated by the user

PRM-DBRC
Permanent error

RBA
Relative Byte Address of the actual database block in error

Following are the fields in Part II, Database Dynamic Backout Errors:

Database
Name of the database

PSB Name
Name of the Program Specification Block needing backout

Date
Julian date the dynamic backout error occurred

Time
Time the dynamic backout error occurred

To limit the scope of the RDBE panel, add a suboperand. Here are two examples:

DISPLAY RDBE (DBXINV02)
Limits scope of display to the databases named DBXINV02.

DISPLAY RDBE (DBX*)
Displays all databases that begin with DBX.

Displaying Unschedulable Transactions Statistics (RTXU panel)

The Unschedulable Transactions Statistics panel shows statistics about the transactions that cannot be scheduled and that are on suspend queues.

The following example of an Unschedulable Transactions Statistics report shows how the system displays IMS statistics that are based on the transaction activity.

```
+================== UNSCHEDULABLE TRANSACTIONS STATISTICS ===================+
| PERIOD:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
|  Transaction   |   PSB Name    |  SERIAL  |  Status   |   COUNT  |    Time    |
+----------------+---------------+----------+-----------+----------+------------+
|  ADFUTOSA      |   ADFUTOSA    |   No     |  Suspend  |     5    |  00:14:00  |
+----------------+---------------+----------+-----------+----------+------------+
|  ADFUT001      |   ADFUT001    |   No     |  Ustopped |    N/A   |  00:04:00  |
+----------------+---------------+----------+-----------+----------+------------+
|  ADFUT00B      |   ADFUT00B    |   Yes    |  Ustopped |    N/A   |  00:20:00  |
+============================================================================+
```

Figure 93. Unschedulable Transactions Statistics report

The following are the fields in the RTXU panel:

Transaction
Name of the unschedulable transaction

PSB Name
Name of the program the transaction uses

SERIAL
The sequence of the transactions, with values of:

Yes
Presents the transactions to the application in the same sequence as the transactions are entered into the IMS system.

No
Default

Status
Current status of the transaction is one of the following states:

Stopped
A /STOP TRANS command is issued against the transaction.

Pstopped
A /PSTOP TRANS command is issued against the transaction.

Ustopped
Indicates one of the following states:

- You assigned SERIAL=Yes to this transaction and IMS cannot process a message for this transaction because of unavailable database resources.
- You assigned SERIAL=No to this transaction and its suspend counter became greater than 10.

Purged
A /PURGE TRANS command is issued against the transaction.

Locked
A /LOCK TRANS command is issued against the transaction.

Suspend
Indicates that there are messages on the transaction's suspend queue because requested database resources are not available. IMS suspends these messages from processing until you issue one of the following commands:

- /STA DB for the unavailable database resource
- /DEQ SUSPEND for the transaction's suspend queue

- /NRE IMS warm start
- /ERE IMS emergency restart

Count
Value of the SUSPEND counter

See the Ustopped status that is previously described. Count applies only to transactions in the SUSPEND state.

Time
Length of time in hours, minutes, and seconds that the transaction is found in this state

To limit the scope of the RTXU display, add a suboperand. Here are two examples:

DISPLAY RTXU (ADDINV)
Limits scope of display to the transactions names ADDINV.

DISPLAY RTXU (ADF*)
Displays all transactions that begin with ADF.

Displaying Fast Path General Information (RFPI panel)

The Fast Path General Information panel provides information on fast path message driven, non-message driven, and utility regions and BALG messages.

The following example of an Fast Path General Information report shows how the system displays IMS statistics based on the fast path message driven, non-message driven, and utility regions.

```
+======================= Fast Path General Information =======================+
| Period:  11:45 to 12:00 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|  Fast Path Message Driven Regions = 2                                       |
|  Fast Path Non-Message Driven Regions = 1                                   |
|  Fast Path Utility Message Regions = 0                                      |
|  BALG Msgs Processed = 2487        BALG Msgs queued = 12                    |
|  BALG Msg Enq Rate = 84/sec        BALG Deq Rate = 84/sec                   |
|  Fast Path trace ACTIVE                                                     |
+=============================================================================+
```

Figure 94. Fast Path General Information report

Following are the fields in the RFPI panel:

Fast Path Message Driven Regions
Average number of Fast Path message driven regions active during this interval.

Fast Path Non-Message Driven Regions
Average number of Fast Path non-message driven regions active during this interval. This count reflects BMPs that are processing Fast Path PSBs.

Fast Path Utility Regions
Average number of Fast Path utility regions active during this interval.

BALG Msgs processed
Total number of Fast Path transactions processed during this interval.

BALG Msgs queued
Average number of Fast Path transactions queued during this interval.

BALG Msg Enq Rate
Fast Path transaction enqueue rate during this interval.

BALG Deq Rate
Fast Path transaction dequeue rate during this interval.

Fast Path trace (active/ inactive)
EPILOG detected that the Fast Path trace was active/inactive sometime during this interval.

Displaying Fast Path Buffer Pool Statistics (RFPP panel)

The Fast Path Pool Statistics panel provides size and usage information on fast path buffer pools.

The following example of a Fast Path Pool Statistics report shows how the system displays IMS statistics based on the fast path buffer pool.

```
+======================= Fast Path Pool Statistics =========================+
| Period:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+---------------------------------------------------------------------------+
|                        Fast Path Buffer Pool Statistics                   |
+---------------------------------------------------------------------------+
| DBBF (Total FP buffers defined)  =   30                                   |
| DBFX (FP buffers for System use) =   10                                   |
| BSIZE (Size of each FP buffer)   = 2048                                   |
| Avg number of regions waiting for free buffers   3                        |
+---------------------------------------------------------------------------+
|                        Fast Path Buffer Pool Usage                        |
+---------------------------------------------------------------------------+
|       |  Total | Unfix Bufs |             Page Fix buffers                |
|       |  DBBF  | New Region |  SDEPS | Region Use |  Avail |  Int Q |  OTHR |
+-------+--------+------------+--------+------------+--------+--------+-------+
|# Bufs |     30 |         11 |      1 |            |     16 |      2 |       |
+-------+--------+------------+--------+------------+--------+--------+-------+
|BSIZE  |   2048 |       2048 |   2048 |       2048 |   2048 |   2048 |  2048 |
+-------+--------+------------+--------+------------+--------+--------+-------+
|Bytes  |  61440 |      22528 |   2048 |            |  32768 |   4096 |       |
+-------+--------+------------+--------+------------+--------+--------+-------+
|Percent|  100.0 |       36.6 |    3.3 |            |   53.3 |    6.6 |       |
+===========================================================================+
```

Figure 95. Fast Path Pool Statistics report

The following are the fields in the RFPP panel:

DBBF
: Total number of Fast Path database buffers defined.

  This is the DBBF parameter in the IMS startup procedure.

DBFX
: Buffers defined for system use.

  This is the DBFX parameter in the IMS startup procedure.

BSIZE
: Size of each Fast Path buffer.

This is the BSIZE parameter in the IMS startup procedure.

Avg number of regions waiting for free buffers
: Average number of regions waiting for a free buffer during this interval.

Total DBBF
: Total buffers in the Fast Path buffer pool.

Unfix Bufs New Region
: Average number of Fast Path buffers defined that are not page fixed for use by regions or output thread use.

  Unfix Bufs New Region is the number of non-fixed buffers that are available for normal buffer allocation (NBA) for additional (new) regions.

SDEPS
: Average number of page fixed buffers that have a sequential dependent control interval (SDEP CI) defined.

Region Use
: Average number of buffers which the dependent regions used for normal buffer allocation and overflow buffer allocation (OBA).

Avail
Average number of buffers found allocated but not used for dependent region NBA or OBA.

Avail also includes buffers are available for system or output thread usage but not currently being used for these purposes, for example, buffers that are not allocated for dependent region NBA or OBA.

Int Q
Average number of buffers found in the intermediate queue.

These are buffers that have been previously used and released, and now await return to the page fixed available allocation.

OTHR
Average number of buffers currently being written back to DASD by output threads

Displaying Fast Path Output Thread Statistics (RFPO panel)

The Fast Path Output Thread Statistics panel provides information on fast path output thread activity.

The following example of an Fast Path Output Thread Statistics report shows how the system displays IMS statistics based on the fast path output thread activity.

```
+==================== Fast Path Output Thread Statistics ===================+
| Period:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+-------------------------------------------------------------------------+
|   OTHREAD Defined  |      Active  |      Idle  |      Wait  |       Queued  |
+--------------------+--------------+------------+------------+---------------+
|                 3  |           2  |         1  |         0  |            0  |
+--------------------+--------------+------------+------------+---------------+
+=========================================================================+
```

Figure 96. Fast Path Output Thread Statistics report

The following are the fields in the RFPO panel:

OTHREAD Defined
Total number of output threads defined

Active
Average number of output threads active during this interval

Idle
Average number of output threads idle during this interval

Wait
Average number of buffers found waiting for an output thread during this interval

Queued
Average number of buffers found queued on an output thread

Displaying Fast Path Region Statistics (RFPR panel)

The Fast Path Region Statistics panel provides statistics on the number of sync points, DEDB and MSDB calls, overflow and normal buffers used, and unit-of-work contentions.

The following example of an Fast Path Region Statistics report shows how the system displays IMS statistics based on the fast path region.

```
+==================== Fast Path Region Statistics ============================+
| Period:  13:00 to 13:15 on 08/29/08                  Elap =15:00 M     IMSD |
+-----------------------------------------------------------------------------+
|        |  SYNC | DEDB | DEDB | MSDB |  OBA |  NBA |  UOW |  CI |  DEDB      |
| Region | points| call | I/O  | call | high | defn | cont | cont|buffer waits|
+--------+-------+------+------+------+------+------+------+-----+------------+
| FPMSG1 |    10 | 1236 |  124 |    0 |    0 |    2 |    0 |   0 |          0 |
| FPMSG2 |     5 |    0 |    0 | 1238 |    0 |    1 |    0 |   0 |          0 |
| FPMSG3 |    30 |52987 | 3527 |    0 |    0 |    1 |   20 |   0 |          4 |
| FPMSG5 |   100 | 120K | 5639 |    0 |    2 |    0 |   87 |   0 |         14 |
+--------+-------+------+------+------+------+------+------+-----+------------+
| Total  |   145 | 174K | 9290 | 1238 |    2 |    4 |  107 |   0 |         18 |
+=============================================================================+
```

Figure 97. Fast Path Region Statistics report

The RFPR data is collected from the IMS X ’5937’ log record.

Following are the fields in the RFPR panel.

Region
Region name.

SYNC points
Number of successfully completed sync points during this interval.

DEDB call
Total number of DEDB calls during this interval.

DEDB I/O
Total number of DEDB I/Os during this interval.

MSDB call
Total number of MSDB calls during this interval.

OBA high
Highest number of overflow buffers used during this interval.

NBA defn
Average number of normal buffers defined for this region during this interval.

This field is obtained from internal IMS control blocks, not from IMS log records.

UOW cont
Number of unit-of-work contentions during this interval.

DEDB buffer waits
Number of waits for a database buffer during this interval.

To limit the scope of the RFPR panel, add a sub-operand. For example:

DISPLAY RFPR (FPMSG3)
Limits scope of display to the region named FPMSG3.

Displaying Receive Any Pool Statistics (RECA panel)

The Receive Any Pool Statistics panel provides size and usage information on the Receive Any pool.

The following example of a Receive Any Pool Statistics report shows how the system displays IMS statistics based on the Receive Any pool.

```
+======================= Receive Any Pool Statistics =======================+
| Period:  13:00 to 13:15 on 08/29/08                   Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
|                           RECA (Receive Any) Pool                          |
+----------------------------------------------------------------------------+
|  Total   |  Util   |  HWM util   | Bufr size |  #RECA |  #RECA use | #RECA HWM |
+---------+--------+------------+----------+--------+------------+-----------+
|  49152   |  83.3% |      91.7%  |      4096 |     12 |          10 |         11 |
+---------+--------+------------+----------+--------+------------+-----------+
+============================================================================+
```

Figure 98. Receive Any Pool Statistics report

The following are the fields in the RECA panel:

Total
Average total size of the receive any pool

Util
Average pool utilization during the collection interval

HWM util
High water mark pool utilization during the collection interval

Bufr size
Average receive any pool buffer size

#RECA
Average number of receive any buffers defined during the collection interval

#RECA use
Average number of receive any buffers used during the collection interval

#RECA HWM
Greatest number of receive any buffers used during the collection interval

Displaying Sequential Buffering Statistics (RBUF panel)

The Sequential Buffering Statistics panel provides sequential buffering storage and usage information.

The following example of a Sequential Buffering Statistics report shows how the system displays IMS statistics based on the buffering storage and usage information.

```
+===================== Sequential Buffering Statistics =====================+
| Period:  13:00 to 13:15 on 08/29/08                   Elap =15:00 M     IMSD |
+----------------------------------------------------------------------------+
|                      Sequential Buffering Storage Usage                      |
+----------------------------------------------------------------------------+
|   Maximum storage    |   Current storage    |   Current util    |   HWM util    |
+---------------------+---------------------+------------------+-------------+
|             76120    |              21346   |            28.0%   |      37.8%    |
+---------------------+---------------------+------------------+-------------+
+============================================================================+
```

Figure 99. Sequential Buffering Statistics report

The following are the fields in the RBUF panel:

Maximum storage
Average amount of maximum storage which could have been used for sequential buffering during the collection interval

Current storage
Average amount of storage used for sequential buffering during the collection interval

Current util
Average percent of storage utilized for sequential buffering during the collection interval

HWM util
High water mark percent of storage used for sequential buffering during the collection interval

Chapter 12. Reference: Bottleneck Analysis execution states

The following topics describe the execution group states and the individual execution states. Competing and non-competing states are mutually exclusive, while executing states are a subset of competing states.

The execution group states are as follows:

- CPU usage
- IMS scheduling waits
- Database I/O waits
- z/OS waits
- IMS waits
- Output waits
- External subsystem (ESS) waits

CPU usage execution states

Bottleneck analysis is a technique that focuses on processor usage. Removing bottlenecks enables transactions to use more processor per unit of time and therefore complete faster. The individual execution states in the CPU Usage group describe how much processor the transactions use and where (example, in DL/I vs. in the application program). All these states are running states and are only experienced by transactions actually running in a dependent region.

You can use the USING CPU states to infer activity which bottleneck analysis cannot measure directly. For example, a shift away from USING CPU towards DATABASE I/O indicates, in the absence of increased I/O contention, that the accessed databases can require reorganization. An abrupt, massive increase in USING CPU across all transaction types is indicative of hardware processor degradation. For example, the cache buffer or the TLB might go out of service and the message that is issued by the system check handler might go unnoticed or might be misunderstood.

Applying the CPU in APPL state

The collector includes a transaction in the USING CPU IN APPL state when it determines that the application program that is processing the transaction is running instructions on a processor. The program had no DL/I function in progress, so the collector ascribes its use of the processor to application processing.

Note, however, that the collector counts CPU used by z/OS components such as program fetch in this category. In a normal IMS system this number is small compared to the other execution states, yet this statistic is one of the most important statistics you can watch because of the following considerations:

- Almost every tuning trick serves to increase this number (and decrease the total of the other execution states by the same amount). Indeed, a transaction that is running 100% in this state is not degraded at all unless it is looping.
- Almost every performance problem serves to decrease this number (and increase the total of the other execution states by the same amount). For example, if I/O contention slows down an application's access to database, its execution profile moves towards DATABASE I/O and away from this execution state.

Applying the CPU in IMS state

The collector determines that the application program that is processing the transaction is running instructions on a processor. The program had a DL/I function in progress, so the collector ascribed its use of the processor to IMS processing.

An example of such processing is the CPU used by DL/I action modules to search a buffer pool for a record before you issue an I/O to retrieve it.

IMS scheduling waits

When IMS schedules transactions, appropriate resources must be available, otherwise the transaction must wait. There are multiple points within scheduling where waiting can occur. Bottleneck analysis shows the dynamic characteristics of transactions that are waiting during the scheduling process.

IMS transactions are serviced by application programs that are running within message processing regions (MPPs) or message-driven batch-message processing regions (BMPs). Each region can service one transaction at a time. When more transactions arrive than the active regions can service, IMS hold them in the message queues until the regions complete current work, or until IMS activates more regions.

IMS incurs a resource usage when it activates an application program within a message region. IMS applications can process multiple transactions each time they are scheduled. The PROCLIM parameter of the TRANSACT macro limits the number of transactions IMS services each time the application is activated.

You can customize IMS applications so that transactions are serviced by only one region, or by multiple regions in parallel. The PARLIM parameter of the TRANSACT macro controls the number of parallel regions. If the number of transactions queued exceeds the PARLIM, then IMS initiates another region to service them.

When a transaction is waiting for IMS to schedule it into a region, bottleneck analysis distinguishes between various cases:

WAIT FOR GU
The number of queued transactions that are not serviced by an active application for one of the following reasons:

- The application might not be active because message processing regions are currently servicing other transactions.
- The number of transactions queued exceeds the cumulative remaining PROCLIM values of the regions currently processing this transaction. The value of PARLIM does allow the application to be scheduled in another message processing region, if one is available.

WAIT FOR MPP
The number of transactions queued that is not serviced by an active application for one of the following reasons:

- The application might not be active because message processing regions are currently servicing other transactions.
- The number of transactions queued exceeds the cumulative remaining PROCLIM values of the regions currently processing this transaction. The value of PARLIM does allow the application to be scheduled in another message processing region, if one is available.

WAIT FOR RESCHEDULE
The number of transactions queued that is not serviced by an active application because the number of transactions queued exceeds the cumulative remaining PROCLIM values of the regions currently processing this transaction. The value of PARLIM does not allow the application to be scheduled in another message processing region, if one is available.

WAIT FOR BMP

The collector finds that the transaction is waiting in the message queue and it is a transaction that is destined for a message-driven batch message processing (BMP) region. There is no reason (example, an unavailable database) why the BMP cannot run at the moment, but it is not running. (The user submits a job to start BMPs; IMS does not automatically schedule them like it schedules MPPs.)

This state is a noncompeting state. It is a situation analogous to a batch job, which is TYPRUN=HOLD. There is nothing IMS or those trying to tune it can do to make such transactions run faster. They cannot run at all until the user chooses to submit the BMP job.

INTENT CONFLICT

The collector finds that the transaction is waiting in the message queue. A message processing region (MPP) attempted to schedule it, but is unable to because the PSB associated with the transaction indicated an intent towards one of the databases it accesses which the PSB cannot commit to at the moment (example, PROCOPT=E when other scheduled PSBs are using the database). Intent conflicts also result from applications that request parallel scheduling when load balancing is disabled at the transaction level.

SCHEDULING BLOCKED

The collector finds that the transaction is waiting in the message queue. There is no reason (example, PSB stopped) why IMS cannot schedule the transaction. At least one MPP sensitive to the class of the transaction type attempted to schedule another transaction type but is halted for some reason (example, intent conflict) and the scheduling options that are specified for that other transaction prevented IMS from giving up on the transaction and possibly attempting to schedule another one. When a scheduling fails for intent, IMS does not attempt to schedule it again until another MPP stops. When IMS does attempt it again, the scheduling rules that are specified in the failing transaction apply.

BLOCK MOVER WAIT

IMS cannot schedule a transaction into a dependent region until IMS loads its PSB and all the DMBs it requires from ACBLIB. IMS allows only one PST to load blocks at a time; any others that want to load blocks are put on a queue. ACBLIB is on a single volume, which can have only one I/O at a time. Serialization is also a result of the SMGT latch. The transaction that is counted here has a PST that is attempting to schedule it, which the collector finds on that queue.

PSB POOL

The collector finds that the transaction is waiting in the message queue. IMS cannot schedule a transaction into a dependent region until its PSB is in the PSB pool. If the PSB is not a RESIDENT one (or even if it is parallel-schedulable), IMS must find room for a copy of the PSB in the pool before scheduling can proceed. If IMS cannot find space, the scheduling attempt fails and the PST remains idle until an application ends. Scheduling then proceeds according to the scheduled option for the failing transaction. The transaction that is counted here is found waiting for space in the PSB pool when a PST attempts to schedule it. As a result, the PST is not available for use by other transactions.

PSBW POOL

The collector finds that the transaction is waiting in the message queue. At least one message processing region is available to service this transaction, but scheduling fails because there is not enough available space in the PSB work pool for this transaction to run. The transaction needs space in the PSB work pool for such things as an I/O area, an area for segment search arguments (SSA), and a copy of the user parameter list.

DMB POOL

A transaction cannot be run in a dependent region until the DMBs for all of the databases it uses are in the DMB pool. For those DMBs that are not RESIDENT, IMS must find room for a copy in the pool before scheduling can complete. If space cannot be found, the PST trying to schedule the transaction IWAITs for it. The transaction that is counted here is found waiting for space in the DMB pool when a PST attempted to schedule it. As a result, the PST is not available for use by other transactions.

Note: Space is freed up in the DMB pool by deleting DMB blocks not currently in use. Before a DMB block can be deleted, all of the data sets it uses must be CLOSEd. The next time that DMB is needed it must be reloaded and its data sets re-OPENed. OPEN and CLOSE halt IMS processing and start z/OS services, and are therefore considered highly undesirable.

BLOCK LOADER

IMS cannot schedule a transaction into a dependent region until its PSB and all the DMBs it requires are loaded from ACBLIB. When the collector sees IMS routines that are working on scheduling a transaction while loading the blocks from ACBLIB, it counts that transaction in this category.

For more information about this execution state, see the sections about the block mover, PSB pool, and DMB pool execution states.

UNSCHEDULABLE

The collector found the transaction is in the message queue and determines that it cannot currently be scheduled, even if there are no other transactions that are locking it.

IMS cannot schedule the transaction for one of the following reasons:

- The transaction type is STOPped.
- The class of the transaction type is USTOPped.
- The program (PSB) which processes the transaction type is STOPped.
- The program (PSB) which processes the transaction type is not found on ACBLIB.
- One or more databases that are required by the PSB that processes the transaction type are unavailable (for example, STOPped, and DMB not found).
- Unschedulable state
- For transactions destined for an MPP, there are no MPPs running that are sensitive to the class of the transaction.
- For transactions destined for an MPP, the current scheduling priority is zero.

The collector can count both MPP and BMP transaction types in this state.

WAIT FOR IFP

This execution state is the number of transactions queued waiting to be serviced by a fast path application program that is running in a fast path Message region. The application is servicing other transactions.

Database I/O Waits

Many database DL/I requests can be satisfied from records already in a buffer pool. Others, however, do require that IMS complete physical I/O. EPILOG shows a subtotal for database I/O, and then if you specified that such statistics be kept by the collector, it shows the breakdown of the I/O by database (no further breakdown, such as by data set group or individual data set is available from bottleneck analysis).

See the DBIO keyword in "EPILOG collector keywords" on page 13 for information about how to do this.

The individual execution states within this category are the DMBnames of each database to which a transaction did I/O.

There is no way to tell from this statistic alone which stages of I/O processing (for example, active on the UCB, waiting in the logical channel queue) the I/O operations for the databases were in. A tool such as DEXAN for MVS can provide more insight.

In addition to IMS/VS full function databases, physical I/O operations that are performed to data entry databases (DEDBs) are also reported. If the statistics are requested by database, the area name are shown as the execution state.

DEDB updates are completed by output threads asynchronously after the transaction completes and the changes are logged. I/O operations that are performed by output threads are not a source of bottleneck for IMS/VS transactions and therefore the transactions are not affected in the database I/O waits.

z/OS Waits

While running in a message processing region or a batch message processing region, a transaction might be blocked by contention for resources that are managed by z/OS (as opposed to resources managed by IMS). An example of such a resource is the processor. The individual execution states in this group show you when and where transactions get slowed down by resources managed by z/OS.

CPU Wait (CTL)

The collector found the transaction ready to run on a CPU within the IMS control region, but all online CPUs in the complex were running some other work. The CPUs might be running other IMS control region tasks, IMS dependent regions, or a task not related to IMS, such as TSO.

The collector does not count itself in this analysis. If the only reason a transaction is not executing on a CPU is that the collector is executing, the collector counts the transaction as USING CPU. This is a standard sampling technique for measuring CPU usage by workload. Also, the collector is unable to see CPU Wait which results from the execution of work with a higher dispatching priority than itself (disabled, global SRB, higher priority address spaces (example, *MASTER*), and local SRBs in the IMS CTL region). If there is much of this activity, the effect will be to skew the measurements. Some amount of CPU Wait and perhaps even other states such as database I/O will be seen as USING CPU.

CPU Wait (DEP)

The collector found the transaction ready to run on a CPU within its dependent region, but all online CPUs in the complex were running some other z/OS address space. The CPUs might be running the IMS control region, other dependent regions, or address spaces not related to IMS such as TSO.

BLDL I/O

When IMS schedules a transaction into a message processing region, its application program must be in virtual storage before it can begin to execute. If the application is not one which is PRELOADed, IMS must invoke z/OS program fetch to bring it into virtual storage from PGMLIB.

The collector considers a transaction to be in this execution state if it observes PDS directory I/O done on behalf of a BLDL request. This occurs if IMS does not find the BLDL entry for the module to load in the MPP's BLDL look-aside buffer (or if z/OS refuses to use the one it finds because PGMLIB is authorized).

Program Fetch I/O

When IMS has scheduled a transaction into a message processing region, its application program must be in virtual storage before it can begin to execute. If the application is not one which is PRELOADed, IMS must invoke z/OS program fetch to bring it into virtual storage from PGMLIB.

The collector considers a transaction to be in this execution state if it observes I/O done by program control interrupt (PCI) fetch. If the module to be loaded is large, program fetch I/O can be very time-consuming.

Application I/O

The collector found the transaction scheduled into an MPP and saw that MPP issued an I/O request which did not go through DL/I. The collector also noticed that the I/O was not of the type used by program fetch. It is usual for installation standards not to allow applications to do I/O of the type this section discusses. This standard might be violated if a debugging feature uses an output file that remains on when the program is put into production.

Swapping In

The collector found the transaction active inside an MPP which was in the process of being swapped in by z/OS. Typically, IMS makes all its dependent regions nonswappable, so some sort of installation modification must be suspected.

CTL Private Page

The collector found the transaction active inside an MPP. he application program processing the transaction was ISWITCHed to the IMS control region and was found waiting for one of the following reasons:

- Its ITASK suffered a page fault for a page in the private area of the control region's address space.
- Its ITASK was ready to use the CPU, but it was waiting on the IMS dispatcher's READY queue while the control task was serialized by a page fault taken in the private area by some other ITASK.

The reason the IMS control region can stand so little paging is that when one ITASK page faults in the CTL region, all ITASKs running in it halt.

CTL EXT PRIV Page

This execution state is the same as the CTL PRIVATE PAGE execution state, except that the private page z/OS referenced is located in the control region's extended private area, which is located above the 16 megabyte addressing line.

DEP Private Page

The collector found the transaction active inside an MPP. The application program processing the transaction (or possibly some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page within its private area.

Dependent external private page

This execution state is the same as the DEP PRIVATE PAGE execution state, except that the private page z/OS referenced is located in the dependent region's extended private area, above the 16 megabyte addressing line.

CSA page

The collector found the transaction active inside an MPP or IFP with its execution blocked for one of the following reasons:

- The application program (or, more likely, some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the common storage area (CSA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault in CSA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in CSA.
- The application program was waiting for the resolution of a page fault for a page in the common storage area (CSA) occupied by a pageable main storage database (MSDB).

ECSA page

The collector found the transaction active inside an MPP or IFP with its execution blocked for one of the following reasons

- The application program (or, more likely, some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the extended common storage area (ECSA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault in extended CSA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in extended CSA.

- The application program was waiting for the resolution of a page fault for a page in the extended common storage area (ECSA) occupied by a pageable MSDB. This can only occur when running IMS release 2.2 or higher.

LPA page
The collector found the transaction active inside an MPP with its execution blocked Execution was blocked for one of two reasons:

- The application program (most likely compiler runtime routines copied to LPA or some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the pageable link pack area (PLPA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While over there, the collector found its ITASK waiting for the resolution of a page fault in LPA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in LPA.

ELPA page
The collector found the transaction active inside an MPP with its execution blocked for one of two reasons:

- The application program (most likely compiler runtime routines copied to LPA or some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the extended pageable link pack area (EPLPA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault in extended LPA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in extended LPA.

Cross memory page
The collector found the transaction active inside an MPP. The IMS system was running with LSO=X. The application program processing the transaction performed a DL/I call which was processed by IMS modules running in the control region. Instead of running in the control task, the task is running within the Address Space Control Block (ASCB)/Task Control Block (TCB) of the dependent region, but in the virtual storage associated with the control region, via the z/OS cross memory services. DL/I processing had suffered a page fault for a page in the control region's private area. In this situation, the dependent region's execution is suspended while the page fault is resolved. Processing by the IMS control task is unaffected.

CTL private page (GFA)
The collector found the transaction active inside an MPP. The application program processing the transaction was ISWITCHed to the IMS control region and one of two things happened:

- Its ITASK had suffered a page fault for a page in the private area of the control region's address space.
- Its ITASK was ready to use the CPU, but it was waiting on the IMS dispatcher's READY queue while the control task was serialized by a page fault taken in the private area by some other ITASK.

In either case, the page fault occurred at a time when the available frame queue (AFQ) was completely empty, so the request for the resolution of the page fault had to be put on a special real storage manager (RSM) queue called the general frame allocation (GFA) queue while the system stealing frames to replenish the AFQ.

As z/OS is shipped, the SRM is set up with an AVQLOW of 10 and an AVQOK of 14. This means that SRM will start stealing frames when there are 10 or fewer on the AFQ, and will steal enough to get the count back to 14. If you consider for a moment that to load a 200K application takes 50 frames, and that (z/OS/370) program fetch asks for them 4 at a time, you will see the exposure an IMS system could have to long waits as SRM made pass after pass collecting the necessary frames.

GFA wait is not a large problem because SRM STEAL is not the only source of replenishment for the AFQ. Every time an address space is physically swapped out, the frames it occupied go on the AFQ. Also, when a FREEMAIN is issued, all frames backing the virtual storage released are made available.

DEP private page (GFA)

The collector found the transaction active inside an MPP. The application program processing the transaction (or possibly some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page within its private area. The request for resolution of the page fault was found on the general frame allocation (GFA) queue, indicating that it had occurred when RSM was completely out of frames (AFQ of 0).

Refer to the discussion of GFA under CTL private page (GFA).

CSA page (GFA)

The collector found the transaction active inside an MPP with its execution blocked for one of two reasons:

- The application program (or, more likely, some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the common storage area (CSA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While over there, the collector found its ITASK waiting for the resolution of a page fault it took in CSA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in CSA.

In either case, the page fault occurred at a time when the available frame queue (AFQ) was completely empty, so the request for the resolution of the page fault had to be put on the general frame allocation (GFA) queue, and that is where the collector found it.

LPA page (GFA)

The collector found the transaction active inside an MPP with its execution blocked or one of two reasons:

- The application program (or, more likely, some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the pageable link pack area (PLPA).
- The application program processing the transaction had ISWITCHed to the IMS control region. While over there, the collector found its ITASK waiting for the resolution of a page fault it took in LPA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in LPA.

In either case, the page fault occurred at a time when the available frame queue (AFQ) was completely empty, so the request for the resolution of the page fault had to be put on the general frame allocation (GFA) queue, and that is where the collector found it.

X-MEM page (GFA)

The collector found the transaction active inside an MPP. The IMS system was running with LSO=X. The application program processing the transaction had done a DL/I call which was being processed by IMS modules running in the control region, but under the ASCB/TCB of the dependent region via z/OS cross memory services. DL/I processing had suffered a page fault for a page in the control region's private area. The collector found the request for resolution of the page fault on the GFA queue, indicating that it had occurred when the AFQ was empty. In this situation, the dependent region's execution is suspended while the page fault is resolved. Processing by the IMS control task is unaffected.

IMS waits

While running in a message processing region or a batch message processing region, a transaction might be blocked by contention for resources that are managed by IMS (as opposed to resources managed by z/OS). An example of such a resource is an IMS latch. The individual execution states in this group show you when and where transactions are slowed down by resources that are managed by IMS.

MFS load
: The collector found an I/O in progress against the message format services (MFS) data set. This means an MFS format block was being loaded so that an input or output message could be formatted by MFS.

SPA I/O
: The collector found the transaction waiting for I/O against the scratch pad area (SPA) data set. This data set is used by conversational transactions which do not use an in-core SPA.

QBLKS I/O
: The collector found the transaction waiting for I/O to complete to the queue blocks data set.

SHMSG I/O
: The collector found the transaction waiting for I/O on the short message (SHMSG) data set.

LGMSG I/O
: The collector found the transaction waiting for I/O on the long message (LGMSG) data set.

PI wait
: The collector found the application program processing the transaction IWAITing for a program isolation (PI) resource. Program isolation is the mechanism which allows databases to be simultaneously accessed and even updated by different users.

  A PI resource is essentially a DMB number to identify the physical database, and an RBA (relative byte address) to identify the physical block being held. When two programs go after the same PI resource at the same time, one must wait.

Sync point wait
: The collector found the transaction waiting for I/O to complete as the result of synchronization (SYNC) point processing.

Other DL/I IWAIT
: The collector found the application program processing the transaction in DL/I somewhere, but it could not ascribe the condition to any of the DL/I oriented states listed previously. Currently, PSTs waiting for space in both the ISAM/OSAM and VSAM pools will have the transactions they are processing counted in this category.

  Note: This execution state should be less than 5%.

IWAIT in IMS dispatcher
: IWAIT in IMS dispatcher is a measure of the IMS dispatching queue for dependent region activities. The collector found the application program processing the transaction IWAITing in the IMS dispatcher somewhere, and could not ascribe the condition to any of the DL/I oriented states listed previously. As the workload rises, this value also rises. This number should be less than 5%.

IWAIT in term
: The collector found the application program processing the transaction IWAITing. It also found an indication that the application program was terminating. The collector could not ascribe the status of the transaction to any of the states listed previously.

  Note: This execution state should be less than 5%. Termination IWAITs may be ascribed to sync point processing of an update transaction. VSAM write BFRs of type=tran are issued under control of the dependent region.

Pseudo abend
The collector found that the application program processing the transaction was IWAITing and terminating due to a pseudo abend.

Other abend
The collector found the application program processing the transaction IWAITing. It also found an indication that the application program was terminating due to an abend or because the dependent region was cancelled.

LOGL latch
The collector found the application program running the transaction waiting for the LOGL latch. A latch is the IMS version of an z/OS suspend lock; it is a fast, cheap way of serializing access to some resource which cannot be used by more than one transaction at a time.

The IMS logical logger uses the LOGL latch to serialize access to the buffers which the physical logger writes out to the log tape. If the physical logger gets behind, ITASKs pile up on this latch.

SMGT latch
The collector found the application program running the transaction waiting for the SMGT (storage management) latch.

The IMS storage management module (DFSISMN0) uses this latch to serialize access to various control blocks and buffers, such as the message queue, CIOP, CWAP, DMB, and PSB pools.

XCNQ latch
The collector found the application program running the transaction waiting for the XCNQ latch.

Program isolation uses the XCNQ latch to serialize access to the PI ENQ/DEQ queues and control blocks.

DLOG latch
The collector found the application program running the transaction waiting for the DLOG latch.

IMS uses the DLOG latch to serialize access to the dynamic log buffers, queues, and tables.

GCMD latch
The collector found the application program running the transaction waiting for the GCMD latch.

The GCMD latch is the global command latch.

AUTH latch
The collector found the application program running the transaction waiting for the AUTH latch.

The AUTH latch is the authorized processing latch.

ACTL latch
The collector found the application program running the transaction waiting for the ACTL (statistics logging) latch.

IMS uses the ACTL latch to serialize access to the DC monitor log buffers and control blocks. When the DC monitor is active, the control region and all parallel DL/I tasks compete for this latch.

GEN1 latch
The collector found the application program running the transaction waiting for a generic latch.

The generic latch is a class of latches which currently consists of the DMBE (data management block) and DBBP (database buffer pool) latches. The latches are generic in that there is one DMBE latch for each DMB defined to the IMS system, and one DBBP latch for each subpool used by the ISAM/OSAM buffer handler.

IENQ latch
The collector found the application program running the transaction waiting for the IENQ latch. The IENQ latch serializes accesses to internal buffers and control blocks within the IMS control region.

HDSM latch
The collector found the application program running the transaction waiting for the HDSM latch. The HDSM latch serializes processing of DL/I HD space management operations; it finds or frees space within an HD database.

QBUF latch
The collector found the application program running the transaction waiting for the QBUF latch. The QBUF latch serializes accesses to message queue buffers, and their associated control information.

OBFM latch
The collector found the application program running the transaction waiting for the OBFM latch. The OBFM latch serializes accesses to OSAM buffers, and their associated control information.

VBFM latch
The collector found the application program running the transaction waiting for the VBFM latch. The VBFM latch serializes accesses to VSAM buffers, and their associated control information.

DBBP latch
The collector found the application program running the transaction waiting for the DBBP latch. The DBBP latch serializes accesses to database buffers, and their associated control blocks.

DMBE latch
The collector found the application program running the transaction waiting for the DMBE latch. The DMBE latch is used to serialize the dynamic insertion and removal of control blocks associated with databases (DMBs).

CBTS latch
The collector found the application program running the transaction waiting for the CBTS latch. IMS uses the CBTS latch to serialize alterations to dynamic control block chains (IPAGES) within dynamic storage management.

NQDQ latch
The collector found the application program running the transaction waiting for the NQDQ latch. The NQDQ latch serializes ENQ/DEQ activity.

SNDQ latch
The collector found the application program running the transaction waiting for the SMB enqueue/dequeue latch. IMS uses the SNDQ latch to serialize the dynamic insertion and removal of scheduling management blocks (SMBs).

Other latch
The collector found the application program running the transaction in ISERWAIT (latch wait), but not waiting for any of the latches previously described.

ISWITCHED to CTL

The collector found the application program running the transaction to have executed an ISWITCH macro. The PST is still executing, but in the control region instead of in the dependent region.

There are a number of conditions under which this occurs. These include, but are not limited to:

- I/O to ISAM files
- Obtaining input messages (including IMS message queue I/O if it is required)
- ISRTing output messages (including IMS message queue I/O if it is required)
- EOV processing on all database data sets (when such is required to free up a buffer)

- Processing required as part of the scheduling process which is not accounted for in the scheduling-related execution states described previously.

ISWITCHED to LSO

The collector found the application program running the transaction to have executed an ISWITCH macro. The PST is still executing, but in the control region instead of in the dependent region. This means instead of running in the control task, the PST is:

- For LSO=Y, running under a subtask in the control region which is reserved for its use only. LSO=Y costs a substantial amount of CPU, but saves virtual storage in CSA at the cost of virtual storage in the private area of the IMS control region.
- For LSO=X, running under the Address Space Control Block (ASCB)/Task Control Block (TCB) of the dependent region, but in the virtual storage associated with the control region. This is done for the same reason as LSO=Y, but costs less CPU because of the greater efficiency of z/OS cross memory services over the old WAIT/POST logic.

Other wait

The collector found a transaction executing, but could not ascribe its execution state to any of the categories previously documented.

Fast Path buffer

The collector found that the application program running the transaction was waiting for a Fast Path buffer. Buffer allocation for the IFP is less than the maximum number of allowable buffers the Normal Buffer Allocation (NBA) execution parameter specifies.

Fast Path overflow buffer lock

The collector found that the application program running the transaction was waiting for the Fast Path overflow buffer lock.

If an IFP attempts to allocate buffers beyond the normal number of buffers allowed (NBA), overflow processing takes place. Overflow processing attempts to increase the buffer allocation by the amount allowed for overflow (OBA). The overflow lock is required to allocate more buffers than the number specified by normal buffer allocation. The overflow lock is enqueued by IMS Fast Path (IFP) and held until sync point processing, at which time the buffer allocation is returned to normal. Overflow interlock is used to serialize overflow buffer allocation.

Fast Path overflow buffer allocation

The collector found that the application program running the transaction was waiting for a Fast Path Overflow Buffer. The IFP reached the maximum number of buffers allowed, and overflow processing is in progress. Overflow processing has increased the buffer allocation by the amount allowed for overflow (OBA).

ISWITCHED to Fast Path TCB

The collector found that the application program running the transaction executed an ISWITCH macro. The PST is still executing, but in the control region under the Fast Path TCB (not in the dependent region).

MSDB latch

The collector found that the application program running the transaction was waiting for the MSDB latch.

Whenever an MSDB call or sync processing routine needs to get or release control of an MSDB resource, the MSDB latch must be obtained.

DMAC latch

The collector found that the application program running the transaction was waiting for the DMAC latch.

The DMAC latch is used to serialize updates to DEDB area data sets. Sync point processing modules require the DMAC latch. Updates are recorded in the log records. The physical update of the DEDB area does not take place until the updates are written to the IMS log.

DMAC share latch

The collector found that the application program running the transaction was waiting for the DMAC share latch.

The DMAC share latch serializes access to DMAC control blocks. Open and close area data set processing modules require the DMAC share latch in exclusive mode.

While a Fast Path message processing region terminates, diagnostic information is copied from the control region to the dependent region. This function requires the DMAC share latch in share mode. Waits only occur for this function because there can be only one requestor of the DMAC share latch in exclusive mode at one time, and the DMAC latch serializes these modules.

ADSC directory latch

The collector found that the application program running the transaction was waiting for the ADSC directory latch.

The ADSC directory latch serializes access to the ADSC directory. This is a wait that uses the dynamic control block latch (CBTS). The ADSC directory latch is obtained in the open and close data entry database modules, and Fast Path command processing modules.

FNCB latch

The collector found that the application program running the transaction was waiting for the FNCB latch.

The FNCB latch serializes access to the notify control block chain. Various IMS commands or options, like opening or closing a DEDB, use the notify facility for communication.

Fast Path resource latch

The collector found that the application program running the transaction was waiting for the Fast Path resource latch.

The Fast Path Resource latch serializes access to the exclusive control resource blocks (XCRBs). XCRBs represent the status of a currently owned (possibly exclusively controlled) resource within a DEDB area.

Fast Path sync processing

The collector found the transaction in sync point processing. During this state the dependent region is using CPU; however, the amount of time spent in this state should be small.

Fast Path sync lock

The collector found the application program running the transactions waiting for the Fast Path sync lock.

The sync lock serializes the sync point processing in IFPs with check-point processing and other activities which stop a DEDB area, such as after you issue a /STOP AREA, /STOP ADS, or /DBR AREA command, or if a physical I/O error is detected.

The sync point processing function requests sync lock in share mode; all other activities require the lock held in exclusive mode.

DEDB area lock

The collector found that the application program running the transaction was waiting for the DEDB area lock.

The DEDB area lock serializes updates to a DEDB area data set. There is a different lock for each area. The DEDB area lock is obtained in sync point processing.

A DEDB area lock is always taken before the DMAC latch to avoid the possibility of a deadlock.

DEDB segment

The collector found that the application program running the transaction was waiting to obtain control of a resource in a data entry database (DEDB). The resource under control is a unit of work (UOW), and it is represented by an XCRB control block.

MSDB segment

The collector found that the application program running the transaction was waiting to obtain control of a resource in a main storage database (MSDB). The resource under control is an MSDB segment.

Fast Path IWAIT in term

The collector found the Fast Path application program processing the transaction IWAITing, indicating that it was terminating.

Fast Path pseudo abend

The collector found the Fast Path application program processing the transaction IWAITing and that the transaction processing in the message region pseudo-abended.

Fast Path other abend

The collector found that the Fast Path application program processing the transaction was IWAITing and that the application program was terminating due to an abend or cancelled dependent region.

Open/Close latch

The collector found that the application program running the transaction was waiting for the open/close latch. The open/close latch serializes resources to Fast Path databases.

IRLM wait

The collector found the application program executing inside a dependent region. The dependent region issued a request to the IRLM address space and was waiting for a response.

An application program will wait until IRLM has granted authorization to a database. This can result in extended response times for those transactions being processed by this application program.

Wait for HSSP pool space

The collector found the application program executing inside a dependent HSSP BMP region. The application program is waiting for space in the HSSP private pool.

High speed sequential processing (HSSP) is a feature that improves the sequential processing of Fast Path data entry databases (DEDBs).

Output waits

When outbound processing of a message is delayed because IMS is waiting for an output-related service or resource, EPILOG detects an output wait.

Message on fast path output queue

The collector might find that an output message, which is generated by the transaction in the output queue, is waiting to be dequeued. The fast path output router, which runs as an IMS task in the control region, dequeues fast path output messages.

Message on non-fast path output queue

The collector might find that an output message, which is generated by a transaction in the non-fast path output queue, is waiting to be dequeued.

External Subsystem Waits

Create thread

The collector finds that the dependent region is creating a thread. A thread is the DB2®® structure that describes an application connection between DB2 and IMS regions.

When a create-thread is in progress, DB2 is allocating an application plan to process the SQL® call. An application plan defines the DB2 resources that are accessed from an application program.

Terminate thread

The collector finds that the dependent region is ending a thread. When a syncpoint is completed, the thread ends and the plan that is used by the SQL call is deallocated unless the region is a WFI. DB2 database might also occur when the thread ends.

Wait for subsystem call

The collector finds that the application program is waiting for DB2 to complete an SQL call.

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.OMEGAMON monitoring products support several user interfaces. Product functionality and accessibility features vary according to the interface.

The major accessibility features in this product enable users in the following ways:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

In addition, the product documentation was modified to include the following features to aid accessibility:

- All documentation is available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Some content presented in IBM Documentation might not yet be in a format that a screen reader can process. If you need help, contact ibmkc@us.ibm.com.

Interface information

The Tivoli Enterprise Portal interface offers the greatest range of functionality, but is not entirely accessible. The OMEGAMON enhanced 3270 user interface offers more limited functionality, but is entirely accessible. (The enhanced 3270 user interface supports all the accessibility features supported by your emulator. If you are using IBM Personal Communications, you can find information about its accessibility features in the Using Emulator Sessions topic. If you are using a third-party emulator, see the documentation for that product for accessibility information.)

The OMEGAMON ("classic") interface uses an ISPF style interface. Standard and custom PF Key settings, menu options, and command-line interface options allow for short cuts to commonly viewed screens. While basic customization options allow for highlights and other eye-catcher techniques to be added to the interface, the customization options are limited.

Related accessibility information

Some content presented in IBM Documentation might not yet be in a format that a screen reader can process. If you need help, contact ibmkc@us.ibm.com.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online
Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.

Troubleshooting Guide
For more information about resolving problems, see the product's Troubleshooting Guide.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

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

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

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM 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 Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

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.

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 actual people or business enterprises is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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" at http://www.ibm.com/legal/copytrade.shtml.

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

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

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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.

Privacy policy considerations

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.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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 http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Special Characters

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z