IBM OMEGAMON for IMS on z/OS
5.5.0

Realtime Commands Reference

IBM

Note

Before using this information and the product it supports, read the information in "Product legal notices" on page 333.

April 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. Operational essentials for your OMEGAMON for IMS system

You can manage the performance of your IMS system by using the user interfaces, product components, modes of operation, and INFO-line format of OMEGAMON® for IMS.

OMEGAMON for IMS provides tools that you can use to manage your IMS system resources effectively, and to assist the help desk staff to identify and resolve problems:

- Performance indicators alert you to impending problems
- You can quickly go to detailed panels for more information about problems
- You can use the Operator Assist function to display database transactions and programs

For example, you can request a view of only those resources that are unavailable. Then, if you see a stopped database, you can move your cursor that is next to the database name and type the action code to start the database. OMEGAMON for IMS creates the appropriate IMS command and passes it to IMS for execution.

Managing IMS performance

Managing IMS performance requires you to balance workloads and resources, and monitor the effects of your load balancing decisions. You monitor IMS to determine whether IMS transactions are processing at an acceptable rate.

Even though IMS is a complex environment, something as simple as a stopped database can delay the completion of the IMS transactions. Typically, users might observe the effects of resource contention or unavailable resources and inform help desk personnel.

Managing your system is an iterative process that comprises the following steps:

1. Monitor the performance of your system.
2. Identify the root cause when a workload or key system resource shows a problem.
3. Correct problems that arise.
4. Use the data from your observations and your knowledge of data center priorities to determine performance targets.
5. Continually monitor your system.

Monitor performance

Your site might not have formal, written performance objectives, but almost all sites have informal objectives. OMEGAMON for IMS can help you translate informal objectives into formal objectives, and can also help you to monitor performance on your system. After OMEGAMON for IMS is up and running, you can start monitoring current IMS performance by asking yourself the following questions:

- How are your groups of transactions performing?
- How are your IMS resources performing?
- Are the problems that OMEGAMON for IMS identifies really problems for my site?
- Is OMEGAMON for IMS missing any problems for my site?

See the *IBM Tivoli OMEGAMON for IMS on z/OS: Response Time Analysis Reference* for more information about how to set response-time exception values.

Identify and correct problems

When one of the status indicators signals that a problem exists, OMEGAMON for IMS can help you identify the cause of this problem.

For example, OMEGAMON for IMS informs you what might be causing poor performance for a group of transactions, and helps you identify which resources or system components are busiest or unusually active when the response time for a group of transactions is delayed. You can use this information to correct the problem. If you must issue an IMS command to correct the problem, you can go to panels where you can select IMS commands from a menu, or you can issue IMS commands directly.

The OMEGAMON for IMS online help system also provides you with context-sensitive help to resolve problems.

User interfaces

With OMEGAMON for IMS, multiple user interfaces provide access to IMS information.

Use the following interfaces:

- By using the easy-to-use menu interface, you can access real-time data where each menu option leads you to a panel that displays appropriate commands and output.
- By using the command interface, you can access a set of extensive and flexible commands that covers almost every aspect of the IMS environment in real time.
- By using the OMEGAMON for IMS historical component, you can access historical information, request printed reports by using a batch report generator, or make interactive queries through a series of TSO panels. For information about these interfaces, see the *IBM Tivoli OMEGAMON on z/OS: Historical Component (EPILOG) User's Guide*.

To go from the menu interface to the command interface or from the command interface to the menu interface, press PF12.

Product components

When you install the IBM® OMEGAMON for IMS on z/OS® product and OMEGAMON for IMS component, more components are integrated with the product. Some components are required for OMEGAMON for IMS to function while other components are optional.

The following table describes the product components that are available at installation and indicates whether the component is required or optional.

Table 1. Product components available with OMEGAMON for IMS

Component	Description
Realtime Monitor component (required)	Provides basic real-time information about the IMS environment
Menu interface for the Realtime Monitor (required)	Provides real-time information about an IMS subsystem by using the original OMEGAMON for IMS menu system interface
Command interface for the Realtime Monitor (required)	Provides real-time information about an IMS subsystem by using an extensive set of flexible commands
OMEGAMON Subsystem (optional)	Provides dynamic I/O information to OMEGAMON for IMS
Response Time Analysis (optional)	Provides monitoring of IMS transaction and end-to-end response time
Application Trace Facility (ATF) (optional)	Tracks activity on a transaction by transaction basis and records the individual events for transactions
Bottleneck Analysis (optional)	Provides information for degradation analysis
Historical component (optional)	Provides historical information about the IMS environment

Table 1. Product components available with OMEGAMON for IMS (continued)	
Component	**Description**
IMS Console Facility (optional)	Provides a complete IMS Master Console for OMEGAMON for IMS

PARMGEN is the method that you use to configure Tivoli® Management Services on z/OS components and OMEGAMON V5.3.0 and higher monitoring agents. For more information about PARMGEN and configuring the components, see the *OMEGAMON shared documentation Version 6.3.0 Fix Pack 2 and above* and the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

Modes of operation

When you configure OMEGAMON for IMS, you are requested to select and customize an operating mode.

Available operating modes are as follows:

- VTAM® mode
- TSO/ISPF mode
- Dedicated mode

The following table describes each operating mode and its requirements.

Table 2. Characteristics and requirements for OMEGAMON for IMS modes of operation		
Mode	**Characteristics**	**Configuration requirements**
VTAM	Use the VTAM mode to run OMEGAMON for IMS sessions from a VTAM terminal without an intermediate online application, such as TSO. You can set automatic update mode so that the screen refreshes automatically. VTAM mode allows all VTAM terminal users to share a single copy of OMEGAMON for IMS.	Define a VTAM applid for OBVTAM.
TSO and ISPF	The TSO address space communicates with the OMEGAMON for IMS address space by using a VTAM application, VTM1. In this mode there, is no auto screen refresh; the screen refreshes when you press the Enter key. Use TSO mode to access OMEGAMON for IMS without logging off TSO. ISPF mode includes split-screen capability that you can swap between multiple OMEGAMON for IMS sessions, or between OMEGAMON for IMS and another ISPF application.	• Define a VTAM applid for OBVTAM. • Requires an active OBVTAM application. • Define a set of virtual terminals to VTAM. You can define up to 99 virtual terminals in the virtual terminal pool (VTPOOL).
Dedicated	Dedicated mode offers high availability and does not require VTAM services. Dedicated mode uses EXCP to communicate with a terminal and refreshes the screen every few seconds. Dedicated mode allows OMEGAMON for IMS to provide real-time data even when VTAM is not available.	Availability of a locally attached non-SNA terminal.

If you choose to log on to the menu and command interfaces directly, you can use several different modes of operation that includes dedicated, VTAM, TSO, and ISPF modes.

INFO-line format

The first line of an OMEGAMON for IMS screen display is called the INFO-line. The INFO-line accepts keyboard input over the underscores, for example the /PRINT command, and displays status information about your session.

The INFO-line looks similar to this output:

```
________________ ZMENU     VTM      OI-II     V530./C I91C 05/11/15 13:42:12     B
```

Figure 1. INFO-line example

The INFO-line is composed of the following areas.

INFO-line format

```
<input>___________ cccccccc mode LOG pr cc sysid mm:dd:yy hh:mm:ss nnn Ac
```

Figure 2. INFO-line format

input area
Accepts INFO-line commands or screen space names. A screen space is a set of commands that is saved under a unique name. Each of the commands on the predefined screen starts when the screen space name is entered. OMEGAMON for IMS distinguishes screen space names from INFO-line commands because INFO-line commands always start with a slash (/). They are explained in the next section.

cccccccc
Shows the name of the screen space currently in use, if any.

mode
Shows the type of session or mode that is used to access OMEGAMON for IMS. The possible values are:

DED
A dedicated mode session.

DIR
A director segment that runs dedicated in a cross memory or cross system mode session.

DSK
A collector segment that runs in a cross system mode session.

VTD
A director segment that runs under VTAM in a cross memory or cross system mode session.

VTM
A VTAM mode session

VTS
An ISPF mode session that runs under VTAM.

VTT
A TSO mode session that runs under VTAM.

XMM
A collector segment that runs in a cross memory mode session.

LOG
If the word LOG displays, the screen is copied to the REPORT log file. If logging is off, the field is blank.

pr
Displays a code or multiple codes for the product that is running and its version.

cc
Identifies the user profile in use.

sysid
: Indicates the IMS system ID value. This value identifies which IMS system OMEGAMON for IMS is monitoring.

mm:dd:yy
: Indicates the date on which the screen was last refreshed.

hh:mm:ss
: Indicates the time when the screen was last refreshed.

nnn
: Indicates the scrolling depth.

A
: Indicates whether the automatic screen facility is active. If it is, an A displays. If it is not, the field is blank.

c
: The variable *c* can be one of these values:

  S
  : Screen spaces are currently stacked.

  B
  : The terminal bell is activated and sounds if exceptions occur.

  Note: If both screen stacking and the bell feature are active, only an *S* is displayed.

Command types and format

If you want to use the command interface, you must know the types of OMEGAMON for IMS commands that are available and the appropriate time and place to enter each command type.

Command types

Figure 3 on page 5 shows the four types of commands that are used in OMEGAMON for IMS and sample output.

```
/PRINT__________ ZMENU    VTM     OI-II    V530./C I91C 05/11/15 13:42:12    B

DISK    VMXA04    VMXA05    VMSP50    VMHP02    OMONVM    DOSTST    DP215R    DOSRES+
dadr       1A0       1A1       1B0       1B1       2A7       2B0       4F1       4F2
.MIN    DADR    DALC    DIO    DIOQ    DOPN    DRES    DSTA    DTYP    DUSR    DVMP  +
```

Figure 3. Four types of commands

The command types are as follows:

INFO-line
: These commands perform control functions such as printing a screen (/PRINT) and stopping your OMEGAMON for IMS session (/STOP). INFO-line commands are run first and, unlike the other types of commands, no longer display as soon as the commands run. Therefore, you cannot save them in a screen space. INFO-line commands always begin with a slash (/), and must be entered on the top line, that is, the line that begins in column 2. If you are running in an automatic update mode, placing the cursor in column 1 on this line pauses updating until you move the cursor.

Major
: These commands select general categories for display, such as system information, resource usage, or storage usage. In the example, the major command DISK produces a list of online disks. You can enter major commands on any line after the INFO-line.

Minor
: Minor commands display detailed information about the category that the major selects. In the example, the minor command DADR displays the unit addresses of the devices that are listed with DISK. You can enter minor commands on any line after the INFO-line, but commands do not run unless they are preceded by the appropriate major command.

Immediate

Immediate commands serve various functions. Some are system monitoring commands, while others give you information about your session or about OMEGAMON for IMS. In the example, the **.MIN** produces a list of all the minors of DISK. Immediate commands can also provide screen and session controls, and often complete the same or similar functions as INFO-Line commands of the same name. You can include control commands as part of a screen space. Enter immediate commands on any line after the INFO-line. It is permissible to enter immediate commands between a major command and one of its minors.

Figure 4 on page 6 is another example of OMEGAMON for IMS commands and command output. The command fields comprise four characters, but you can enter commands with labels and arguments. The format is described in Table 3 on page 6

```
 .MJ DS DISK  DLST  DSKB  DSKC  DSKE  DSKG  DSKM  DSKN  DSKP  DSKQ  DSKR DSKS+

 DISK    VMXA04   VMXA05   VMSP50   VMHP02   OMONVM   DOSTST   DP215R   DOSRES+
 dadr       1A0      1A1      1B0      1B1      2A7      2B0      4F1      4F2

2GDEV  SYSDA
         MIS003  HSM001  WORKB1  PDB002  MIS002  MIS003  SYSB24  SYSD22 PROD02 +
```

Figure 4. Example of commands

Command format

Many commands can be entered with labels and arguments which can change the output that displays. The following table identifies the constituent parts of the `lccccnn` command:

Table 3. `lccccnn` command entry area

Field	Column	Description
l	1	Label field Many commands accept a character in this field that alters the type of output displayed. The field is also used for special command modifiers, such as those that request online help text for commands. In the example in Figure 4 on page 6 (since the GDEV command produces output that extends for several lines), the numeral 2 in the label field skips the first two rows of available output.
cccc	2 - 5	Command name This field contains the command name and is four characters in length. Although the command name field is columns 2 through 5, OMEGAMON for IMS detects most commands that begin in column 1 and automatically moves the command one space to the right.
nn	6 - 7	Argument field Many commands accept arguments that modify their function or specify output options. In the example in Figure 4 on page 6, the `.MJ` command produces a list of major commands. The command has an argument of DS, a group code, which specifies that only disk major commands be listed.

Table 3. lccccnn command entry area (continued)

Field	Column	Description
	8 - 72	Extended argument Arguments are entered in columns 8 through 72. In the example in Figure 4 on page 6, the GDEV command has an argument of SYSDA, which is a generic device name in the system. Thus, only devices in the SYSDA category are displayed. Many commands also accept keywords and parameters that can extend to column 72. A parameter string cannot be extended onto a second line. The command (along with keywords) must be reentered on succeeding lines

Major, minor, and immediate command help

The OMEGAMON for IMS product provides online help for every type of command. You can access help descriptions for major, minor, or immediate commands.

You can use one of three symbols in column 1 in front of a major, minor, or immediate command to request an explanation of the command function. All commands offer a brief, one-line help. Many commands also have an extended help, which gives you more information about the command or displays the command operands. A continuation character (+) following the end of the one-line help indicates that more (extended) help is available.

HELP symbols

?
Displays a one-line help statement that stays on the screen until you clear it.

/
Displays an extended help statement that stays on the screen until you clear it.

You can clear the help text with the **.DCL** command.

;
Displays an extended help that no longer displays from the screen on the next cycle.

Help examples

The following examples illustrate how the three help symbols are used with the .WAT command.

This entry for the .WAT immediate command

```
?.WAT
```

produces a one-line help as shown in Figure 5 on page 7.

```
______________________   VTX OIDIRIEI /C  IMSA 08/29/08 10:32:22
>.WAT
>       Specifies a delay before executing the commands that follow. +
```

Figure 5. One-line help

When you use a forward slash instead of a question mark,

```
/.WAT
```

an extended help is displayed as shown in Figure 6 on page 8.

```
>.WAT
>       Specifies a delay before executing the commands that follow.         +
>
>       Type of Command:  Immediate
>
>       Command Format:  .WATnn
>
>        1 | 2    | Definition of operands:
>       -------------------------------------------------------------------
>          | .WAT | nn Specifies the length of the delay (seconds) before
>          |      |      the command(s) that follow are executed.
>
>       EXAMPLE:
>        .WAT02
>        .FGO MYSCREEN
>
>       The commands shown above cause OMEGAMON to wait 2 seconds before
>       fetching screen space MYSCREEN.
```

Figure 6. Extended help

The entry

```
;.WAT
```

produces the same output as the/ `.WAT` command, except that plus signs display in column 1 below the one-line help. The plus signs are continuation symbols and indicate that the extended help text is not displayed on the next cycle.

You can obtain help for a minor command by using the symbols that are previously described, without knowing the associated major command name. In some cases, however, the same minor name is used for multiple majors. If the function of the minor command is the same for all of its associated majors, OMEGAMON for IMS displays a help screen similar to the one in Figure 7 on page 8.

```
?dio
>   DIO is a minor command of the following major(s):
>       DEV   DEVL  DEVP  DISK  DSKA  DSKB  DSKC  DSKE  DSKG  DSKM  DSKN  DSKP
>       DSKQ  DSKR  DSKS  DSKU  DSKV  GDEV  TAPE  TPAL  TPBS  TPCU  TPFR  TPMT
>       TPOF  TP16  TP38  TP62  TP7T  TP80  2305  3330  3340  3350  3375  3380
>   Help:
>       Displays the I/O count since IPL of the device(s).                   +
```

Figure 7. Help screen

In cases where the same minor command name is used for multiple major commands, but the function and help text differs, OMEGAMON for IMS lists the possibilities and instructs you to place your cursor after the required major command name. When you press ENTER, the required help displays.

If you know that the minor command is shared among several major commands, you can type the required major command name on the same line after the help request. For example, the entry

```
/dio DISK
```

produces the help text for the `DIO` minor specific to the `DISK` major command.

INFO-line command help

You can access INFO-line command help by using the `.ILC` immediate command.

INFO-line command help is obtained with the `.ILC` immediate command rather than with a symbol in column 1. Noting that the forward slash is not needed, enter the name of the INFO- line command (*cccccc*) in this format:

```
.ILC cccccc
```

For example, to get help for the /DEF command, enter:

```
.ILC DEF
```

The help output that displays is similar to the output shown in Figure 8 on page 9.

```
>.ILC /DEF
>       Sets definition mode.                                                +
>
>       Type of Command:  INFO-line
>
>       Command Format:  /DEF cccc
>
>        1 | 2       Definition of operands:
>        ---------------------------------------------------------------------
>          | /DEF  <cccc>  ON    Turns on definition mode.  Definition mode
>          |                     suspends command execution (except for the
>          |                     commands that control screen editing
>          |                     functions) so that you can define a screen
>          |                     space without executing commands.  Once you
>          |                     set definition mode with /DEF ON, it remains
>          |                     in effect until you issue /DEF OFF or the
>          |                     screen space is saved or replaced.
>          |               OFF   Restores normal command execution (cancels
>          |                     /DEF ON or /DEF HOLD).
>          |               HOLD  Same as ON argument, but definition mode
>          |                     remains in effect after you save a screen
>          |                     space.  It is only cancelled when you issue
>          |                     /DEF OFF.  Use this option when you want to
>          |                     save 2 or more screens in a row without
>          |                     turning on definition mode each time.
```

Figure 8. INFO-line command help

Chapter 2. Operational commands for your OMEGAMON for IMS system

You can use INFO-line and immediate commands, variables, operands, and PF keys to control the operation of OMEGAMON for IMS.

- "Command modifiers and help symbols" on page 11
- "Operational INFO-line and immediate commands" on page 12

Command modifiers and help symbols

Command modifiers control the function and display of output of commands.

You can use the following generic command modifiers with all commands of the specified types.

```
[n cccc]
[< cccc]
[# cccc]
[> cccc]
[? cccc]
[/ cccc]
[; cccc]
```

n

Controls the display of major command output. When a major command selects a list of items (for example, all online disks), the list might extend to more than one line. The value of *n* is a number from 1 to 9 or a letter from A to Z (representing 10 - 35) that specifies the number of lines to skip from the last line that displays. For example, entering 3DISK on the first row of output from the DISK major command displays the fourth row of output.

<

Causes a major command to display a complete list of selected items, even if the list extends to more than one line.

#

Displays the number of items available for a major command to select.

>

Indicates a comment line. OMEGAMON for IMS generates this character in some commands after they execute to prevent them from running again on succeeding cycles. The character is also displayed in the help text. You can use this character when you create screen spaces to designate the line as comment text. The _ (underscore) is an alias of >.

?

Provides a one-line explanation of the command. If a plus sign (+) displays in column 79 of the one-line explanation, it indicates that a more detailed explanation is available by using the slash or semicolon.

/

Provides a detailed explanation of the command and its syntax. The help stays on the screen until you clear it.

;

Provides a detailed explanation of the command and its syntax. The expanded help no longer displays on the screen after the next cycle.

Rate and difference arguments in minor OMEGAMON for IMS commands

For any minor OMEGAMON for IMS command that normally displays a numeric value, you can add an argument that calculates the rate of a system event during an OMEGAMON for IMS interval or the difference from one interval to the next.

Statistics that show rates or differences are often more useful that the original output of the command. For example, the I/O rate is more meaningful to performance analysis than the raw number of I/Os processed.

For commands that normally display time values, such as processor time, the rate that is calculated represents a scaled percentage of usage (expressed as a decimal value) where the real time of the interval equals 100%.

OMEGAMON for IMS requires data from two cycles to calculate a rate or a difference. On the first cycle (the initialization cycle), a row of eight periods (**........**) displays. The rate displays on the next cycle.

The rate and difference arguments are not effective in the following cases:

- If a major command selects different items from cycle to cycle. For example, the DSKB major command might select different busy disks at each screen update. When a rate argument is used with a minor of DSKB, the periods might display after each update, indicating that OMEGAMON for IMS is initializing each time.
- If a command displays a value that is reset to zero by the system between OMEGAMON for IMS intervals.

  The arguments that are shown in the following example must be entered in columns 6 and 7 and must be entered after a minor command that displays a numeric value.

  ```
       cccc [.D|.R|.H|.M|.S]
  ```

.D
: Causes OMEGAMON for IMS to calculate the difference between the two most recent values of an event that is monitored by the specified command (*cccc*).

.R
: Causes OMEGAMON for IMS to calculate the rate per second of an event that is monitored by the specified command (*cccc*) during the last OMEGAMON for IMS interval. .S is an alias for .R.

.H
: Displays a rate per hour.

.M
: Displays a rate per minute.

.S
: Displays a rate per second.

Operational INFO-line and immediate commands

Use INFO-line and immediate commands to control the operation of OMEGAMON for IMS.

Some of the commands that control the operation of OMEGAMON for IMS are shown in the following list. They are displayed in alphabetical order, starting with special characters.

..bb
: Clears rest of screen (where b indicates a blank)).

  Type:
  : Immediate

  This command clears the entire display down to the end of the logical screen.

====
: Draws a separator line across the screen.

Type:
Immediate

Format:
c====aa

c
Indicates, for terminals that support an extended data stream, the color of the separator line. Replace the variable with the first letter of one of the seven extended colors (Red, Blue, White, Green, Pink, Yellow, or Turquoise). For four- or non-color terminals, OMEGAMON for IMS translates an entry of G or B to low intensity. All other color codes translate to high intensity.

aa
Any two characters you want to use for the separator line. When you invoke this command, the specified characters are repeated across the screen.

For example, the command R====++ creates a line of red pluses across the screen.

```
====++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

/ABORT

Stops processing for a cross memory (XMF) collector immediately.

Type:
INFO-line

If the collector is hung and cannot process a STOP command, you can enter the /ABORT command on the INFO-line of the collector.

This command also frees up any resources that the director associated with it.

CAUTION: Use this command only in an emergency, as it does not bring the collector down cleanly. (If the collector later starts after an /ABORT, it might crash with an unexpected program check.)

See the discussion of XMF in the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

/ATTACH

Attaches the specified cross system session (alias is /A).

Type:
INFO-line

Format:
/ATTACH cccc

The variable *cccc* is the four-character system ID of the cross system collector. For example, the following command attaches cross system collector A083:

```
/ATTACH A083
```

For information about the cross system collector, see the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

/ATTN

Emulates the PA1 (program attention) key.

Type:
INFO-line

/ATTN clears the screen, resets your internal security authorization to the lowest level, returns to default basic color settings, and turns off extended color.

/AUPON

Enables automatic update mode.

Type:
INFO-line

This command is valid only in VTAM mode. Automatic update mode is similar to running OMEGAMON for IMS in dedicated mode, since the screen updates at regular intervals without pressing **Enter**. You can check the current default interval or change it with the .SET command.

While OMEGAMON for IMS is automatically updating in VTAM mode, you can continue to enter commands. OMEGAMON for IMS delays input processing in order to avoid executing half-entered input. The length of the delay is determined by the IODELAY keyword of the .SET command.

Important: Some network programs do not support automatic update mode (for example, a program that emulates a terminal for your OMEGAMON for IMS VTAM mode session).

.AUP is the equivalent immediate command.

/AUPOFF

Disables automatic update mode.

Type:

INFO-line

This command turns off the automatic update mode. See /AUPON for additional information about automatic update mode.

Important: Some network programs do not support automatic update mode (for example, a program that emulates a terminal for your OMEGAMON for IMS VTAM mode session).

.AUP is the equivalent immediate command.

.AUP

Controls automatic update mode.

Type:

Immediate

See /AUPON, the equivalent INFO-line command, for the command description. If .AUP is entered without an argument, OMEGAMON for IMS displays whether automatic update mode is ON or OFF.

.CN

Controls the specified secondary console in dedicated mode.

Type:

Immediate

Format:

.CNxxxx or .CN cc

The variable *xxxx* is the hexadecimal address of the secondary console. In dedicated mode, you can set up a secondary OMEGAMON for IMS console to be used for output only. The secondary console is a repeater console; it echoes everything that displays on the primary console, but accepts no commands or input of any kind.

After you set the address of the secondary console with .CNxxxx, you can manipulate it with the following arguments:

.CN OP

Allocate (open) a secondary console for OMEGAMON for IMS output display.

.CN CL

Deallocate (close) a secondary console.

.CN SW

Swap primary and secondary console functions.

The secondary console must be the same terminal type as the primary console.

.D

Deletes the specified number of lines on the physical screen.

Type:

Immediate

Format:
.Dnn

This command deletes lines (beginning with the current line). For example, the following command deletes five lines on the physical screen.

```
.D5
```

/DCL
Deletes all comment lines on the screen.

Type:
INFO-line

If you want to delete only those comment lines after a certain point on the screen, use the equivalent immediate command instead.

.DCL
Deletes all comment lines after its entry line.

Type:
Immediate

If you want to delete all comment lines on the screen, use the equivalent INFO-line command instead. Unlike most other immediate commands, .DCL no longer displays after it executes.

.DD
Deletes a block of data.

Type:
Immediate

To delete a block of data from the physical screen, enter **.DD** on the first line of the block and **.DD** on the last line. For example, the following command deletes the line with the first .DD command and the succeeding 3 lines. **.DD** must be followed by a blank.

```
.DD
DISK   SYSB24   TSO021   SYSB21   z/OSA21
DSKB   z/OSA21   PROD05   SYSA24
.DD 150 334     D8B
```

/DEF
Inhibits automatic updating to allow screen space definition.

Use this command to define a screen space to include commands that comment themselves out or otherwise change form after execution.

Type:
INFO-line

Format:
/DEF {ON|HOLD|OFF}

ON
/DEF ON inhibits automatic updating during a dedicated mode session or a VTAM mode session with automatic updating activated (see the /AUPON, /AUPOFF, or .AUP command). Once you set definition mode with /DEF ON, it remains in effect until you issue /DEF OFF, or save or replace the screen space.

HOLD
Same as ON argument, but definition mode remains in effect after you save a screen space. It is only cancelled when you issue /DEF OFF. Use this option when you want to save two or more screens in a row without reactivating definition mode each time.

OFF
Restores normal screen updating (cancels the effect of /DEF ON or /DEF HOLD).

.DEF is the equivalent immediate command.

.DEF

Inhibits automatic updating to allow screen space definition.

Type:
Immediate

See /DEF, the equivalent INFO-line command, for the command description. If .DEF is entered without an argument, OMEGAMON for IMS displays the current definition mode status (ON, OFF, or HOLD).

DELT

Deletes a screen space from main storage and/or from the user's screen space library (RKOIPCSV).

DELT does not delete screen spaces from the IBM-supplied screen space library, which is referenced by the RKOIPCSV DD statement.

Type:
Immediate

Format:
DELTc aaaaaaaaa

c
One of the following arguments that specifies the location of the screen space. Enter it in column 6:

B or blank
Deletes from both main storage and RKOIPCSV (default).

I
Deletes from main storage (in-storage) only.

D
Deletes from RKOIPCSV only.

aaaaaaaa
The screen space name that you want to delete. Specify the name (starting in column 8).

For example, the following command deletes screen space SAMPLE from main storage.

```
DELTI SAMPLE
```

DING

Forces the terminal bell (audible signal) to sound.

Type:
Immediate

The bell must be activated with the BELL=YES option of the OPTN immediate command. You can set the BELL=YES option through the menu system and save your setting in a user profile.

.DIR

Executes a cross memory (XMF) or cross system (XSF) director command.

Type:
Immediate

Format:
.DIR cccccc

Use the .DIR command to issue commands (*cccccc*) that control director and collector functions. Use this capability to execute these commands from a screen space.

The commands are as follows:

```
.DIR ABORT (see /ABORT)
.DIR ATTACH cccc (see /ATTACH)
.DIR GIVE nn cccc (see /GIVE)
.DIR TAKE nn cccc (see /TAKE)
```

/DOWN

Scrolls down the specified number of lines (alias is /D).

Type:
INFO-line

Format:
/DOWN cccc

cccc can be:

nnn
Scrolls *nnn* lines (1 - 999).

BOT
Scrolls to the last logical row.

CSR
Scrolls according to the current location of the cursor. If the cursor is on the INFO-line, the scroll amount is a page.

MAX
Scrolls down the number of LROWS defined for your terminal.

PAGE
By default, the page scrolls down so that the current cursor position is at the end of the physical screen.

/DOWN works only if the number of logical rows is defined to a number greater than the number of physical rows on this terminal. This definition can be changed with the LROWS startup parameter.

If you assign the /DOWN command to a PF key (the default is PF20), you can type any of the optional arguments on the INFO-line before you press the PF key. OMEGAMON for IMS interprets the entry as if you had typed the command plus the argument.

.DSE

Displays the status of stacked screens.

Type:
Immediate

The .DSE command displays the status of screens stacked with the /STK INFO-line command. The information includes the screen space name, the GETMAINed size in bytes of each screen space, a time stamp that indicates when you stacked the screen, the total amount of storage allocated for all stacked screens, and the relative position of the current stack entry pointer.

The current stack entry pointer is the arrow that is labelled **current** in the .DSE display. The entry pointer indicates which screen space in the stack has most recently been referenced by a /STK INFO-line command. If you issue the /STK command with an up or down argument, the pointer moves to the entry before or after the current entry.

Following is a typical .DSE immediate command display.

```
 .DSE            Entry    Screen   Size (bytes)    Time
+                  1     #01          17987      10:27:14
+      current --> 2     SYSLOAD      17987      11:08:30
+                  3     OIINITZZ     17987      11:56:00
+                Total stack size:    53961
```

Figure 9. .DSE immediate command display

.EXM

Lists and executes all minor commands for the preceding major command.

Type:
Immediate

Format:
.EXM [nn|c1 c2]

(blank)
Without operands, the .EXM command lists and executes all the minors.

nn
Skips *nn* minor commands and executes the rest.

c1 c2
Executes all the minor commands that begin with the specified character string or are in the range specified (*c1* to *c2*). A character string can be 1 to 4 characters long.

This command applies only to the major command that immediately precedes it. The .EXM command executes the minors in alphabetical order and shows the number of minors it executed. You can use operands to limit the execution to specified minors.

For example, the following .EXM command executes minors of the DISK command that have names that start with A through F.

```
DISK
.EXM A F
```

The .EXM command comments itself out after it executes.

.FGO
Allows fast access to screen space *cccccccc*.

Type:
Immediate

Format:

```
n.FGO cccccccc

        [CPSER  {=|EQ|GE|GT|LE|LT|NE} argument]
        [DIR    {=|EQ|GE|GT|LE|LT|NE} argument]
        [MODE   {=|EQ|GE|GT|LE|LT|NE} argument]
        [OPSYS  {=|EQ|GE|GT|LE|LT|NE} argument]
        [PREFIX {=|EQ|GE|GT|LE|LT|NE} argument]
        [IMSID  {=|EQ|GE|GT|LE|LT|NE} argument]
        [UNIT   {=|EQ|GE|GT|LE|LT|NE} argument]
        [USER   {=|EQ|GE|GT|LE|LT|NE} argument]
        [&var   {=|EQ|GE|GT|LE|LT|NE} argument]
```

Figure 10. .FGO command

n
Use the variable *n*, which is an optional numeric label, to delay the fetch of screen space *cccccccc* for a number of cycles up to 35. Use the numbers 1 - 9 or the letters A - Z (representing 10 - 35 cycles). Each time the screen updates, *n* decrements by 1. When n=0, screen *cccccccc* is fetched on the next cycle.

cccccccc
The variable *cccccccc* specifies the name of a screen space.

The .FGO (Fast GO) command is used when creating screen spaces to fetch the next screen space of a series. It allows screen spaces to be chained together and to execute quickly, bypassing the screen display and the normal OMEGAMON for IMS interval. It is useful in exception analysis for implementing the Automatic Screen Facility (ASF) or the Timed Screen Facility (TSF).

Conditional screen fetching

The .FGO command has a conditional screen fetch feature that fetches a screen space only if a condition is true. The format for a condition is `keyword operator argument`.

Keywords
The following keywords are available for conditional fetching of screen spaces. Their values are initialized by OMEGAMON for IMS.

CPSER
CPU serial number. In a multi-processor environment, the supplied CPU serial number is compared with the serial numbers of all processors in the complex. If the relational argument is equal (= or EQ), OMEGAMON for IMS fetches the screen space the first time it finds a match. If the relational argument is NE, OMEGAMON for IMS fetches the screen space only after it checks all of the processors in the complex.

DIR
The ID assigned to the director in cross system mode.

MODE
The 3-character code for OMEGAMON for IMS's mode of operation. It is displayed on the INFO-line during a session. Refer to "INFO-line format" on page 4 for a list.

OPSYS
The z/OS operating system level (for example, 210 or 310).

PREFIX
The OMEGAMON for IMS product code (for example, OI).

IMSID
The system ID from the SYS= startup parameter.

UNIT
The device number from the UNIT= startup parameter (the primary OMEGAMON for IMS console).

USER
The user profile suffix from the USER= startup parameter.

&var
You can set any comparison that you want. The keyword can be any variable name that is set with the **.VAR** command or any OMEGAMON for IMS-defined variable. The **.VAR** command lists OMEGAMON for IMS-defined variables.

Operators
The following relational operators are available for conditional fetching of screen spaces. The operators require blanks on either side, with the exception of the equal sign (=).

= or EQ
Keyword equals argument.

GE
Keyword is greater than or equal to argument.

GT
Keyword is greater than argument.

LE
Keyword is less than or equal to argument.

LT
Keyword is less than argument.

NE
Keyword is not equal to argument.

argument
The argument is a one- to eight-character value to which OMEGAMON for IMS compares the keyword. The argument can be any variable name that is set with the **.VAR** command or any OMEGAMON for IMS-defined variable. The **.VAR** command lists OMEGAMON for IMS-defined variables.

To protect against the possibility of a looping condition that is caused by the **.FGO** command, OMEGAMON for IMS limits the number of consecutive fetches allowed. The limit is controlled with the FGOLIMIT keyword of **.SET**, which is set to 64 by default (in the profile that IBM provides). After the limit is reached, the **.FGO** command acts like the **.SGO** (Screen Go) command so that screen spaces continue to run, but now they display on each cycle.

Therefore, if .FGO screens spaces cause a loop, you must correct the condition and enable the **.FGO** command again with the FGOLOOP keyword of the **.SET** command.

Because FGOLOOP=ON causes the **.FGO** command to display running screen spaces, you might want to turn it on yourself to test your screen space fetch routines. If multiple **.FGO** commands display on one screen, the last **.FGO** command without a condition, or for which the condition is true, runs.

Example

To fetch screen space SAMPLE only if the terminal address is 05E1, enter:

```
.FGO SAMPLE UNIT=05E1
```

or

```
.FGO SAMPLE UNIT EQ 05E1
```

/GIVE

Gives the specified number of screen rows to the cross memory (XMF) or cross system (XSF) collector (alias is /G).

This command determines the number of lines on the physical screen to be used by a cross memory or cross system collector.

Type:
INFO-line

Format:
/GIVE nn cccc

nn
The number of screen rows. If you omit *nn*, all lines are given to collector *cccc*.

cccc
The XMF or XSF collector to receive the lines.

If you issue this command from a collector without specifying *cccc*, the **/GIVE** command returns *nn* lines to the director's screen segment.

For example, this next command assigns 15 lines to the screen segment for the collector that runs on CPU ID A083.

```
/GIVE 15 A083
```

/HELP

Describes HELP facilities (alias is /H).

Type:
INFO-line

You can use the help screen space to find out more about the functions, features, and operation of OMEGAMON for IMS. For help with an individual major, minor, or immediate command, type a question mark (**?**) in column 1 in front of the command. For help with an INFO-line command, refer to the **.ILC** immediate command.

/HOLD

Controls the execution of OMEGAMON for IMS commands.

Type:
INFO-line

Format:
/HOLD ccc

ON
Suspends OMEGAMON for IMS command execution.

OFF
Returns to normal OMEGAMON for IMS command execution.

Use the **`/HOLD ON`** command to stop automatic updates until you enter **`/HOLD OFF`**. The **`/HOLD`** command is designed for users of VM/PASSTHRU. If you are not a user of VM/PASSTHRU, you can also stop automatic updating by placing the cursor in column 1, row 1.

.I
Inserts *nn* blank lines.

Type:
Immediate

Format:
.I nn

This command inserts *nn* blank lines so that you can insert new commands on the screen. The number of logical rows on your terminal is the maximum.

For example, the next command inserts five lines before the current line.

```
.I  5
```

.ILC
Displays INFO-line commands or their help text.

Type:
Immediate

Format:
.ILC /cccccc

The variable */cccccc* is an INFO-line command name (slash is optional). To display all of the INFO-line commands and their aliases, enter the **`.ILC`** command without a command name. To display help text for a specific INFO-line command, enter the **`.ILC`** command followed by the command.

For example, the next command generates an explanation of the**`/STOP`** INFO-line command.

```
.ILC /STOP
```

/LOG
Sends the current OMEGAMON for IMS REPORT log or the XLFLOG to the printer.

Type:
INFO-line

The **`.LOG`** command is the equivalent immediate command, which additionally offers the PUSH and POP arguments. **`/O`** is an alias for **`/LOG`** with the OUT argument.

.LOG
Sends the current OMEGAMON for IMS REPORT log or the XLFLOG to the printer, or manipulates the status of the log.

Type:
Immediate

Format:
.LOGcccc

.LOG accepts the following arguments.

ON
Starts logging

OFF
Stops logging.

OUT
Prints the current log and leaves it open. The command comments itself out to prevent the log from automatically resetting again on the next cycle.

PUSH
Saves the status of the log (ON or OFF) so that it can be restored when you run the **.LOGPOP** command. Use this capability to manipulate the status of the log in screen spaces invoked by .FGO or .SGO, then return it to its original state after these screen routines are complete. One of the following messages is displayed on the same line as the command.

```
>> Log inactive. Status saved. <<
```

or

```
>> Log active. Status saved. <<
```

POP
Restores the log to the status that is in effect when you run the last .LOGPUSH. One of the following messages is displayed on the same line as the command.

```
>> Log status restored to inactive. <<
```

or

```
>> Log status restored to active. <<
```

Note: The Automatic Screen Facility (ASF) and the Timed Screen Facility (TSF) PUSH and POP automatically.

The log is activated and deactivated with the LOG keyword of the OPTN command. Report characteristics are set with the OUTP major command and its minors. For a full description of the logging facility, see the *IBM OMEGAMON for IMS on z/OS: OMEGAMON for IMS User's Guide*. /LOG is the equivalent INFO-line command. The command, however, does not offer the PUSH and POP arguments.

LSCR
Loads screen space members from the screen space library to main storage.

You can make screen spaces more available and more easily fetched by loading them into main storage with LSCR. For example, if a disk is not available, you can continue to invoke the screen spaces that you loaded into main storage with LSCR.

Type:
Immediate

Format:
LSCR cccccccc cccccccc ... cccccccc

The variables *cccccccc* are screen space names. Specify screen space member names (starting in column 8). You can load as many members as can fit on the input line.

For example, the next command asks OMEGAMON for IMS to load screen spaces ZZ1, ZZ2, and ZZ3 from the screen space library CANSOI to main storage.

```
LSCR  ZZ1 ZZ2 ZZ3
```

.MIN
Lists all minor commands for the preceding major command.

This command applies only to the major command that immediately precedes it. The **.MIN** command displays the minors in alphabetical order.

Type:
Immediate

Format:

[H].MIN [nn|c1 c2]

(blank)
Without operands, the **.MIN** command displays all minor commands of the major.

H
The optional label H displays one-line help information for each of the minor commands. Use the following arguments allow you to limit the help display and avoid scrolling down to see the required help.

n
Skips the first *nn* minor commands.

c1 c2
Specifies a single character string or a range of minors from *c1* to *c2* for the help display. A character string can be 1 - 4 characters long.

The next example displays all minor commands of the **DISK** major command.

```
DISK
.MIN
```

.MIN comments itself out after execution.

.MJ

Lists all major and immediate OMEGAMON for IMS commands.

Type:
Immediate

Format:
H.MJ cc

The optional label *H* displays a one-line help text for each command. The variable *cc* specifies one of the following optional two-character group names.

cc
Command Group

(blank)
All major and immediate commands

CL
Class commands

CV
Conversation commands

DB
DMB commands

DD
Data set commands

DG
DEXAN for IMS commands (when DEXAN is installed)

DM
Display IMS control block commands

DS
Disk commands

DV
Device commands

DY
Dynamic allocation/deallocation commands

ES
External subsystem commands

EX
Exception analysis commands

FP
Fast path commands

FR
Fast path message rate commands

LG
Logging commands

LN
Line commands

ME
Memory commands

MF
MFS-related commands

MI
Miscellaneous commands

NO
Node commands

PB
PSB commands

PD
Dynamic control block commands

PL
Pool commands

PS
Plotting commands

PT
Physical terminal (PTERM) commands

RG
Region commands

RT
RTA commands

TA
Tape commands

TC
TCO/TCF commands

TR
Logical terminal commands

TX
Transaction commands

XR
Transaction rate commands

For example, the following command lists all device major and immediate commands.

```
.MJ DV
```

.MJC
Lists all major commands.

Type:
Immediate

Format:
H.MJCcc

The optional label *H* displays a one-line help text for each command. The variable *cc* is an optional group name. See the .MJ command for a list of these groups.

.MJI
Lists all immediate commands.

Type:
Immediate

Format:
H.MJIcc

The optional label *H* displays a one-line help text for each command. The variable *cc* is an optional group name. See the .MJ command for a list of these groups.

.MMA
Lists all major commands for a minor command.

Type:
Immediate

Format:
.MMA cccc

The variable *cccc* is a minor command.

.MOD
Shows OMEGAMON for IMS module names and addresses.

Type:
Immediate

Format:
.MODc

The optional suffix A (.MODA) lists the module names in alphabetical order. This command provides debugging information, including module names and start addresses. If OMEGAMON for IMS detects a program check, these names and addresses are useful to IBM Software Support.

.NXE
Controls display of exceptions.

Type:
Immediate

Format:
.NXEccc

ON
By default, exceptions are displayed.

OFF
Does not display exceptions.

Enter the .NXE immediate command without an argument to show the status of the exceptions display. The frequency with which an exception runs is controlled by the EXNCYC (EXecute Next CYCle) keyword of the **XACB** command. When you set the EXNCYC parameter to check the exception less often than every OMEGAMON for IMS cycle, the **.NXE** command controls whether the XIMS command continues to display tripped exceptions on the cycles when they are not due for execution.

/O
Prints the existing OMEGAMON for IMS REPORT log without closing the log.

Type:
INFO-line

The **/O** command is an alias for /LOG with the OUT argument. See the description of the **/LOG** or **.LOG** commands for complete information about this command.

.PCS
Displays OMEGAMON for IMS program check statistics.

Type:
Immediate

If OMEGAMON for IMS detects a program check, the information is useful to IBM Software Support for debugging.

.PFK
Displays or resets command mode PF key definitions for the current session.

Type:
Immediate

Format:
c.PFKnn=aaaaaaaa/*bbb...b

(blank)
Without operands, the **.PFK** command displays all current PF key settings for command mode. PF keys without assignments do not display on the screen. The default PF key definitions in command mode differ from those definitions in the menu system.

c
Label E for redefining several PF keys at one time.

nn
PF key number.

aaaaaaaa
Screen space name (1 - 8 characters) or INFO-line command (/*cccccc*).

/*bbb...
Comment of up to 32 characters (bbb...) following the slash and asterisk (/*).

You can define up to 99 physical and logical PF keys. Enter the .PFK command and type the new definition after the equal sign. Type comment text following a slash and asterisk (/*).

For example, the following command sets PF15 to issue the /STOP INFO-line command for this session.

```
.PFK15=/STOP         /* Stops OMEGAMON for IMS
```

Use the same format to assign screen space names to PF keys. For example, the following command sets PF26 to call the screen space DISKS for this session.

```
.PFK26=DISKS         /* DASD information
```

To call a screen space that is assigned to a PF key, press the associated PF key or type its number on the INFO-line.

To delete a definition, enter a single underscore (_) for the definition. For example, the following command deletes the definition for PF18.

```
.PFK18=_
```

Use the following steps to redefine several PF keys at one time without having to retype the .PFK command for each one.

1. Enter E.PFK.

OMEGAMON for IMS gives you an extended display of all current PF key assignments and inserts .PFK before each key number as shown here.

```
+.PFK11=/ZOOM            /* ZOOMING FEATURE
```

2. For each new assignment, blank out the plus sign (+) in front of .PFK and type the new assignment after the equal sign.
3. Press **Enter**.

   The assignments that you entered remain in effect during the session.

Important: At startup, OMEGAMON for IMS runs the screen spaces that contain default PF key assignments. The PF keys that you define with the **.PFK** command are in effect only for that OMEGAMON for IMS session. To make these assignments permanent, you must change the default settings in the screen spaces. For command mode, these screen spaces are KOI@PDEF (PF keys 1 - 12) and KOI@PDF2 (PF keys 13 - 24). For menu mode, the screen spaces are KOI@PNEW (PF keys 1 - 12) and KOI@P2 (PF Keys 13 - 24).

/PRINT

Prints the current logical screen (alias is /P).

Type:
INFO-line

When the screen prints, a **>LOGGED<** message displays on the INFO-line.

Note: The page limit that is set with the **.SET** command does not affect the **/PRINT** command.

.PRM

Displays current values of the OMEGAMON for IMS startup command parameters.

Type:
Immediate

This command displays applicable OMEGAMON for IMS startup parameters in the following order

IOMODE=cc
Current OMEGAMON for IMS I/O mode, which is the two-character code entered as the mode in the startup parameters.

SYS=cccc
Current OMEGAMON for IMS system ID, which is the same system ID that displays on the INFO-line.

DIR=cccc
Director system ID. This ID displays only when the current OMEGAMON for IMS is in collector mode.

USER=cc
User profile suffix, which is the same suffix that displays on the INFO-line.

ROWS=nn
Number of rows on the physical terminal.

LROWS=nnn
Number of logical rows for the output area.

COLS=nnn
Number of columns on the physical terminal.

UNIT=cuu
Terminal address of a dedicated OMEGAMON for IMS session.

.PRT

Prints the specified portion of the screen to the REPORT file.

Type:
Immediate

Format:
.PRTc

Without an argument, .PRT prints a screen image from the INFO-line to the line that contains the **.PRT** command. After the partial screen prints, the **.PRT** changes to a comment. The optional argument H (.PRTH) prevents the **.PRT** command from commenting itself out so that it logs these lines continually.

/PWD

Specifies an OMEGAMON for IMS password or reauthorizes a session.

Type:
INFO-line

Authorized commands require entry of a password for execution. You can use the/PWD command in the following ways.

- To authorize your session for internal security, enter the /PWD command on the INFO-line.

  ```
  /PWD____________
  ```

  The system prompts you for a password.

  ```
  ________________ Enter Password
  ```

  The password does not display on the screen as you type it. It remains in effect until you reset it.
- To reset the security level to 0 after your authorized session, type **/PWD** on the INFO-line, but instead of entering a password, press Enter.

  Authorization is cleared.
- The /PWD command can be entered with your user ID to perform the following functions.
  - Log on to an existing OMEGAMON for IMS session and reauthorize external security to your level for the session.
  - In dedicated mode, gain access to external security.

    When you use the/PWD command with your user ID to log on to an existing session, you cannot change or update your password.

  For details about OMEGAMON for IMS's security features, see the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

.R

Repeats the last major command with all following minors *nnn* times.

Type:
Immediate

Format:
.Rnnn

The .R command repeats all lines from the last major command *nnn* times, as if you repeatedly enter the major followed by its minors. The maximum value of *nnn* is the number of LROWs on your terminal.

The .R command is similar to .RC, which repeats the last major command with all following minor commands as many times as necessary to display all items selected by the major. However, the **.Rnnn** command specifies exactly how many times you want the major command repeated. The .R command is most useful as an editing command when you are creating screen spaces.

The .R command is similar to the other OMEGAMON for IMS line commands .D, .I) in that if you enter it at the beginning of a line of data, that line moves down. Therefore, you do not have to insert a blank line to use this command.

.RC

Repeats the last major command and all following minors until all available output is displayed.

Type:
Immediate

When you enter a major command that selects multiple items, such as disks, only one line of output displays. The .RC command automatically repeats a major command and any minor command until all available lines of output are displayed. Enter .RC after the major and any required minors as shown in the following example:

```
DISK
dadr
dsta
.RC
```

When you press Enter, all online disks are displayed (DISK) along with their unit addresses (DADR) and their mount status (DSTA). The display looks similar to the following output:

```
DISK    VMXA14   VMXA12   OMON28   OMON29   VMXA10   SYSB21   VMXA09   OMSTG1+
dadr       140      141      142      143      144      145      146      147
dsta   Private  Private  Storage  Storage  Private  Private  Private  Private
DISK    PPSMPE   TSO026   QM0001   IMS100   MTLIB3   DB2002   PPSMP3   TSO022+
dadr       148      149      14A      14B      14C      14D      14E      14F
dsta   Private  Storage  Private  Private  Private  Private  Private  Storage
              .                 .                   .                 .
              .                 .                   .                 .
              .                 .                   .                 .
DISK    COM001   COM002   COM003   MP310A   PROD10   PROD16   PROD11
dadr       D89      D8A      D8B      D8C      D8D      D8E      D8F
dsta   Private  Private  Private  Private  Storage  Private  Private
.RC
```

Figure 11. DISK display

RENM
Renames a screen space.

RENM renames screen spaces in main storage or in the user-defined screen space library pointed to by ddname RKOIPCSV. It does not rename IBM-provided screen spaces in the library pointed to by ddname RKOIPROC.

Type:
Immediate

Format:
RENMc oldname newname

c
Enter one of the following arguments in column 6 to specify the location of the screen space.

B or blank
By default, renames in both main storage and RKOIPCSV

D
Rename in RKOIPCSV only.

I
Rename in main storage (in-storage) only.

/REP
Replaces the existing saved screen space of the same name.

Use the /REP command in place of the /SAVE command if you want to replace an existing saved screen space in the user-defined screen space library pointed to by ddname RKOIPCSV.

Type:
INFO-line

Format:
/REP cccccccc,a

cccccccc
Specifies the screen space name (1 - 8 characters)

a
One of three arguments that might follow the screen space name. The argument is separated from the screen space name with a comma (,).

B
By default, replaces in both main storage and RKOIPCSV

D
Replaces in RKOIPCSV only.

I
Replaces in main storage (in-storage) only.

The following example replaces the current screen space SAMPLE with the newly defined screen space in both main storage and RKOIPCSV.

```
/REP SAMPLE,B
```

.REP
Displays and sets printer characteristics for the REPORT file.

Type:
Immediate

When you enter the `.REP` command, a series of keywords displays. You can use these keywords to set printer characteristics. If you change more parameters than fits on one line, use the `OUTP` command instead. The keywords for the `.REP` command (and the `.XLG` command) duplicate the minors of the `OUTP` major command.

When you change any of the parameters (except FOLD) and press Enter, OMEGAMON for IMS automatically spins off the REPORT file and reallocates a new one.

SYSOUT=
SYSOUT class.

HOLD=
Specifies whether output is to be placed in the hold queue.

COPIES=
Specifies the number of copies to print.

FORMS=
Specifies the form on which to print.

DEST=
Destination, user ID, or both (separated by a period, colon, or slash) to receive report.

FOLD=
Folds lowercase characters to uppercase.

The following keywords also display if the default values are modified previously. If not, you can type them in and define a new value.

ID1=
Requests OMEGAMON for IMS session-produced separator pages and page headers. The argument for ID1 can be:

*****
By default, OMEGAMON for IMS generates separator pages and page headers with the appropriate job name printed in block letters on the pages.

NONE
OMEGAMON for IMS does not generate separator pages and page headers.

cccccccc
OMEGAMON for IMS generates separator pages and page headers with *cccccccc* printed in block letters on the pages. *cccccccc* is up to eight user-defined characters.

ID2=
Defines up to 16 characters on the left of the separator page.

ID3=
Defines up to 16 characters in the center of the separator page.

ID4=
Defines up to 16 characters on the right of the separator page.

DDNAME=
Overrides standard OMEGAMON for IMS ddnames.

If you change any parameter other than HOLD=, OMEGAMON for IMS spins off the XLFLOG file and creates a new one. If SYSOUT is active, then ddname is inactive and vice versa. The following table shows the parameters in effect and the default settings for SYSOUT or ddname.

Table 4. Default settings for SYSOUT or ddname

Parameters	Default Values	
	SYSOUT	DDNAME
SYSOUT=	A	(inactive)
HOLD=	NO	(inactive)
COPIES=	1	(inactive)
FORMS=	*NONE*	(inactive)
DEST=	*NONE*	(inactive)
DDNAME=	(inactive)	OIREPORT
FOLD=	YES	YES
ID1=	jobname	jobname
ID2=	(blank)	(blank)
ID3=	(blank)	(blank)
ID4=	(blank)	(blank)

/RESHOW
Reshows the previously saved version of the current screen space.

Type:
INFO-line

The /RESHOW command, set to a PF key, gives you the convenience of refreshing your original screen space with a single keystroke after you make temporary alterations.

.RMF
Displays RMF information.

Type:
Immediate

This command displays the RMF version and version code, the current interval length, and the cycle time.

.RMFS
Changes the RMF level that is set by OMEGAMON for IMS.

Type:
Immediate

Format:
RMFS nnn

nnn specifies the RMF level for your system. For example, enter **RMFS 351** to monitor a system that is running RMF level 3.5.1.

OMEGAMON for IMS now sets the RMF level dynamically when you initialize it. However, OMEGAMON for IMS cannot correctly set the RMF level if RMF is not running, or if OMEGAMON for IMS is not APF-authorized. OMEGAMON for IMS therefore selects a default RMF setting. If this setting is not correct, you can change it with RMFS.

RMFS cannot change the RMF level setting when OMEGAMON for IMS dynamically determines the correct level.

.RTN

Ends an ASF or TSF sequence and returns to the calling screen space.

The .RTN command is required at the end of the last screen space in an Automatic Screen Facility (ASF) or Timed Screen Facility (TSF) sequence. It returns to the calling screen space and re-enables exception analysis for further automatic calls. See the ASF and TSF feature descriptions in Chapter 14, “OMEGAMON for IMS automation and logging features,” on page 317.

Type:
Immediate

Format:
n.RTNcc aaaaaaaa

n

The optional label *n* specifies the number of cycles to delay the return to the calling screen space. The value of *n* can be the numbers 1 - 9 or the letters A - Z (representing 10 - 35). Each time the screen updates, *n* decrements by 1. When *n*=0, the current screen executes and OMEGAMON for IMS fetches the next screen space.

For example, the following command returns to the calling screen space after seven cycles.

```
6.RTN
```

cc

The variable *cc* is the NR (no reset) argument. It prevents the .RTN command from automatically resetting the automatic update interval and the log status.

aaaaaaaa

This optional argument specifies a screen space for ASF or TSF to return to other than the calling screen space.

The .RTN command automatically resets the automatic update interval and the log status to that in effect when the ASF or TSF sequence began.

/SAVE

Saves the specified new screen space (alias is /S).

Type:
INFO-line

Format:
/SAVE cccccccc,a

cccccccc

Specifies the screen space name (1 - 8 characters). The screen space name must be a unique alphanumeric name. It must begin with a character and can contain national characters ($, *, or &). If the name you want to assign exists, use /REP instead of /SAVE.

a

Specifies one of three arguments that might follow the screen space name. The argument is separated from the screen space with a comma (,).

B

Saves the screen space to both RKOIPCSV and main storage (in-storage screen facility).

D or b

Saves the screen space to RKOIPCSV only (default).

I

Saves the screen space to main storage only.

Note: Because screen spaces are saved to the library pointed to by ddname RKOIPCSV, the /SAVE command works only if RKOIPCSV is created and properly concatenated at installation. If you are unable to save your screen space, or if your screen space is successfully saved but OMEGAMON for IMS does not access it, check with your installer or check the *IBM Tivoli OMEGAMON for IMS on z/OS: Planning and Configuration Guide* for details about RKOIPCSV.

This next example saves the current screen space SAMPLE in both main storage and RKOIPCSV.

```
/SAVE SAMPLE,B
```

For guidelines on creating screen spaces, see the *IBM OMEGAMON for IMS on z/OS: OMEGAMON for IMS User's Guide*.

SCRN

Lists screen space member names.

Type:

Immediate

Format:

SCRNc aa bb

c

Enter one of the following arguments in column 6 to specify the source of the member list.

B or b

Lists all screen spaces in both the screen space libraries and main storage (default).

D

Lists all screen spaces in the screen space libraries only.

I

Lists all screen spaces in main storage (in-storage) only.

aa bb

Lists all screen spaces that begin with characters *aa* to *bb* (start *aa* in column 8).

a*

Lists all screen spaces that begin with characters *a* (start *a* in column 8).

The following example lists all screens that have names that begin with *C* or *D*.

```
SCRN C D
```

The next example lists all screen spaces in main storage from PA to PA999999.

```
SCRNI PA*
```

or

```
SCRNI PA  PA
```

.SGO

Fetches the specified screen space on the next cycle.

Type:
Immediate

Format:

```
n.SGO cccccccc

      [CPSER  {=|EQ|GE|GT|LE|LT|NE} argument]
      [DIR    {=|EQ|GE|GT|LE|LT|NE} argument]
      [MODE   {=|EQ|GE|GT|LE|LT|NE} argument]
      [OPSYS  {=|EQ|GE|GT|LE|LT|NE} argument]
      [PREFIX {=|EQ|GE|GT|LE|LT|NE} argument]
      [IMSID  {=|EQ|GE|GT|LE|LT|NE} argument]
      [UNIT   {=|EQ|GE|GT|LE|LT|NE} argument]
      [USER   {=|EQ|GE|GT|LE|LT|NE} argument]
      [&var   {=|EQ|GE|GT|LE|LT|NE} argument]
```

Figure 12. .SGO screen space

n
Use the variable *n*, which is an optional numeric label, to delay the fetch of screen space *cccccccc* for a number of cycles up to 35. Use the numbers 1 - 9 or the letters A - Z (representing 10 - 35 cycles). Each time the screen updates, *n* decrements by 1. When n=0, screen *cccccccc* is fetched on the next cycle.

cccccccc
Specifies the screen space name.

Use the .SGO (Screen GO) command when you create screen spaces to build a series of screen spaces that run in sequence. The .SGO command causes screen spaces to branch to other screen spaces. It is useful for implementing the Automatic Screen Facility (ASF) or the Timed Screen Facility (TSF) features of exception analysis.

Conditional screen fetching

.SGO has a conditional screen fetch feature that fetches a screen space only if a condition is true. The format for a condition is `keyword operator argument`.

Keywords
The following keywords are available for conditional fetching of screen spaces. Their values are initialized by OMEGAMON for IMS.

CPSER
CPU serial number. In a multi-processor environment, the supplied CPU serial number is compared with the serial numbers of all processors in the complex. If the relational argument is equal (= or EQ), OMEGAMON for IMS fetches the screen space the first time it finds a match. If the relational argument is NE, OMEGAMON for IMS fetches the screen space only after it checks all of the processors in the complex.

DIR
The ID assigned to the director in cross system mode.

MODE
The 3-character code for OMEGAMON for IMS's mode of operation. It is displayed on the INFO-line during a session. Refer to "INFO-line format" on page 4 for a list.

OPSYS
The z/OS operating system level (for example, 210 or 310).

PREFIX
The OMEGAMON for IMS product code (for example, OI).

IMSID
The system ID from the SYS= startup parameter.

UNIT
The device number from the UNIT= startup parameter (the primary OMEGAMON for IMS console).

USER
The user profile suffix from the USER= startup parameter.

&var
You can set any comparison that you want. The keyword can be any variable name that is set with the **.VAR** command or any OMEGAMON for IMS-defined variable. The **.VAR** command lists OMEGAMON for IMS-defined variables.

Operators
The following relational operators are available for conditional fetching of screen spaces. The operators require blanks on either side, with the exception of the equal sign (=).

= or EQ
Keyword equals argument.

GE
Keyword is greater than or equal to argument.

GT
Keyword is greater than argument.

LE
Keyword is less than or equal to argument.

LT
Keyword is less than argument.

NE
Keyword is not equal to argument.

argument
The argument is a one- to eight-character value to which OMEGAMON for IMS compares the keyword. The argument can be any variable name that is set with the **.VAR** command or any OMEGAMON for IMS-defined variable. The **.VAR** command lists OMEGAMON for IMS-defined variables.

If multiple **.SGO** commands display on one screen, the last **.SGO** command without a condition, or for which the condition is true, runs.

You can also use the **.FGO** command to fetch screen spaces. **.FGO** functions the same as .SGO except .FGO bypasses the screen display and the OMEGAMON for IMS cycle wait.

Example

To fetch screen space DISK on the next cycle, enter the following.

```
 .SGO DISK
```

To delay the fetch of screen space DISK for 11 cycles, and fetch it on the next cycle after n=0, enter the following.

```
B.SGO DISK
```

To fetch screen space SAMPLE only if you are running in an XA environment, enter the following.

```
 .SGO SAMPLE OPSYS=210
```

or

```
 .SGO SAMPLE OPSYS EQ 210
```

.SPT
Assigns a number to a pattern value to accommodate generic selection of certain major command output.

Type:
Immediate

.SPT can set up to 10 patterns for use with certain major commands to select output that matches the selection pattern. To start a pattern, enter the major command name followed by a slash and the pattern number.

Format:
.SPT/n cccccccc

n
Specifies the pattern number. It can be a number 1 - 9. To set or display the default pattern 0, omit the */n*. The argument */n* can also be:

/D
Displays all the patterns.

/C
Clears all the patterns at once. OMEGAMON for IMS comments out .SPT/C so that the pattern does not run again.

..
(Two periods). Clears the contents of the default pattern.

cccccccc
Specifies the pattern value. You can use an asterisk (*) as a wildcard.

The following example sets pattern 2 to all names that begin with OP.

```
.SPT/2  OP*
```

The following entry starts pattern 2 for the major commands DEVP.

```
DEVP/2
```

/STK
Saves and stacks the current screen output for later recall.

OMEGAMON for IMS currently refreshes the screen every cycle. There are times, however, when you might want to save a screen's output so you can return to it later. You might want to investigate a problem by going to another screen or by issuing commands, and then returning to the original screen. You can do this with the /STK command.

Type:
INFO-line

Format:
/STK ccccc

ccccc
One of the following arguments for recalling and deleting entries from the stack.

(blank)
Without an argument, the **/STK** saves all data on the current screen (including all LROWS), not just the display window visible on the terminal. The maximum number of screens you can stack is 999.

n
Recalls stacked entry *n*.

U (up)
Recalls the entry before the current one. In the .DSE display, the Up argument moves the pointer to the entry above the current pointer.

D (down)
Recalls the entry immediately following the current entry. In the .DSE display, the Down argument moves the pointer to the entry below the current pointer.

EMPTY
Clears the entire contents of the stack.

DEL n
Deletes entry *n*.

R
Recalls the current entry.

The recall functions *n*, U, D, and R do not delete screens from the stack.

The .DSE immediate command displays the status of stacked screens and the amount of storage that is used to stack them. A current entry pointer indicates the most recently referenced screen in the stack.

OMEGAMON for IMS displays a message on the INFO-line when it recalls a screen from the stack to indicate that this screen is not run. If OMEGAMON for IMS recalls a stacked screen while in auto-update mode, it places the recalled screen in HOLD mode until you press ENTER.

The screen stacking feature works with extended color if extended color is on when the screen space is saved and if extended color is on when OMEGAMON for IMS recalls the screen. (See the .SCC command in the Profile menu to activate extended color.)

The following figure shows how the /STK command saves the current screen, M110, onto the stack.

```
/STK____________ M110     VTM     OI-II    V530./C IMSA 05/25/15 10:41:11    S
```

Figure 13. /STK command screen

When there are stacked screens, the INFO-line of any non-stacked screen displays an S on the right of the screen. When the bell is on and a B is displayed, the S overlays the B.

The next figure shows how the /STK command recalls entry number 2 (screen space DISKS) from the stack onto the screen.

```
/STK 2__________ #03      VTM     OI-II    V530./C IMSA 05/25/15 10:41:31    S
```

Figure 14. /STK command screen 2

The following figure shows the output from running /STK 2.

```
________________ <== OB0226: DISKS    recalled                   10:41:34
DISK    SYSB24   WORKB2   z/OSA21   OMON11
dadr       140      142      143      146
dalc                         18       82
dio        311      138    39176    37902
dsta   PRIVATE  STORAGE  PRIVATE  STORAGE
dtyp      3380     3380     3380     3380
```

Figure 15. /STK 2 result

STOP
Stops OMEGAMON for IMS.

Type:
Immediate

The STOP command can be entered either on the INFO-line (/STOP) or in the main body of the screen to stop OMEGAMON for IMS.

/TAKE
Takes the specified number of screen rows from a cross memory (XMF) or cross system (XSF) collector (alias is /T).

Type:
INFO-line

Format:
/TAKE nn cccc

nn
The number of lines that the cross system/cross memory segment takes from segment *cccc*.

If you issue this command from a collector and do not specify *cccc*, it takes *nn* lines from the director's screen segment. If *nn* is omitted, OMEGAMON for IMS takes all lines but one from the specified segment.

For example, the following command takes 15 lines from the screen segment for collector A083:

```
/TAKE  15  A083
```

/TOP
Scrolls to the beginning of the logical screen.

Type:
INFO-line

/UP
Scrolls up the specified number of lines (alias is /U).

Type:
INFO-line

Format:
/UP cccc

cccc can be:

nnn
Scrolls *nnn* lines (from 1 to 999).

CSR
Scrolls according to the current location of the cursor. If the cursor is on the INFO-line, the scroll amount is a page.

MAX
Scrolls to the beginning of the screen.

PAGE
By default, scrolls so that the current cursor position is at the top of the screen.

TOP
Scrolls to the beginning of the screen.

The following example scrolls up 20 lines.

```
/UP 20
```

If you assign the/UP command to a PF key (the default is PF19), you can type any of the optional arguments on the INFO-line before you press the PF key, and OMEGAMON for IMS interprets the entry as if you typed the command plus the arguments.

The/UP command works only if the number of logical rows (LROWS) is defined to a number greater than the number of physical rows on the terminal. This definition can be changed with the LROWS startup parameter.

.VAR
Sets, displays, or deletes variables.

Type:
Immediate

Format:

```
[C].VAR
       {SET  &variable value (comment)}
       {LIST (&variable)}
       {DEL &variable}
       [CPSER  {=|EQ|GE|GT|LE|LT|NE} argument]
       [DIR    {=|EQ|GE|GT|LE|LT|NE} argument]
       [MODE   {=|EQ|GE|GT|LE|LT|NE} argument]
       [OPSYS  {=|EQ|GE|GT|LE|LT|NE} argument]
       [PREFIX {=|EQ|GE|GT|LE|LT|NE} argument]
       [IMSID  {=|EQ|GE|GT|LE|LT|NE} argument]
       [UNIT   {=|EQ|GE|GT|LE|LT|NE} argument]
       [USER   {=|EQ|GE|GT|LE|LT|NE} argument]
       [&var   {=|EQ|GE|GT|LE|LT|NE} argument]
```

Figure 16. .VAR command

C
Requests conditional processing. You can specify a value for a variable when the criteria is matched.

SET or S
Sets or changes the value of a variable.

&variable
The variable must be 1 - 8 characters. The ampersand (&) preceding the variable name is optional.

value
1- to 64-character alphanumeric string assigned to &variable. Single quotation marks are required only if special characters or blanks are used in the string.

comment
1- to 35-character alphanumeric annotation that follows the variable string.

LIST or L
Displays all existing variables or a specified variable. If you omit &variable, OMEGAMON for IMS lists all variables.

DEL
Deletes the specified variable.

Conditional setting of variables

The .VAR command has a conditional feature that sets a variable only if a condition is true. To conditionally set a variable, you specify the *C* label and provide a condition in the `keyword operator argument` format.

Keywords
The following keywords are available for conditional setting of variables. Their values are initialized by OMEGAMON for IMS.

CPSER
CPU serial number. In a multi-processor environment, the supplied CPU serial number is compared with the serial numbers of all processors in the complex. If the relational argument is equal (= or EQ), OMEGAMON for IMS sets the variable the first time it finds a match. If the relational argument is NE, OMEGAMON for IMS sets the variable only after it checks all of the processors in the complex.

DIR
The ID assigned to the director in cross system mode.

MODE
The 3-character code for OMEGAMON for IMS's mode of operation. It is displayed on the INFO-line during a session. Refer to "INFO-line format" on page 4 for a list.

OPSYS
The z/OS operating system level (for example, 210 or 310).

PREFIX
The OMEGAMON for IMS product code (for example, OI).

IMSID
The system ID from the SYS= startup parameter.

UNIT
The device number from the UNIT= startup parameter (the primary OMEGAMON for IMS console).

USER
The user profile suffix from the USER= startup parameter.

&var
You can set any comparison that you want. The keyword can be any variable name that was set with the **.VAR** command or any OMEGAMON for IMS-defined variable.

Operators
The following relational operators are available for conditional setting of variables. The operators require blanks on either side, with the exception of the equal sign (=).

= or EQ
Keyword equals argument.

GE
Keyword is greater than or equal to argument.

GT
Keyword is greater than argument.

LE
Keyword is less than or equal to argument.

LT
Keyword is less than argument.

NE
Keyword is not equal to argument.

argument
The argument is a one- to eight-character value to which OMEGAMON for IMS compares the keyword. The argument can be any variable name that is set with the **.VAR** command or any OMEGAMON for IMS-defined variable.

OMEGAMON for IMS-defined variables

OMEGAMON for IMS predefines the following variables for your use.

- **&ZFRSTSS** - First screen space name.
- /ZOOM INFO-line command variables:

  &ZOOM
  Data that is found at the cursor location

  &ZOOMC
  Command or exception name field

  &ZOOMS
  Originating screen space

- Exception variables, where *cccc* is the exception name:

  &ZXccccT
  Threshold value

  &ZXccccV
  Last trip value

  &ZXccccW
  Worst trip value

Note: The letter *Z* is reserved for IBM use as the first character of a variable.

Setting variables examples

You can use OMEGAMON for IMS variables to build generic screen spaces, pass values to other screen spaces, and alter the flow of screen spaces (`.SGO`).

In the following example, values are set for NXTSCRN and DEVICE.

```
.VAR SET &NXTSCRN MONITOR
.VAR SET &DEVICE 123
```

You can now use those variables in a screen space.

```
DEV &DEVICE
DIO
.SGO &NXTSCRN
```

OMEGAMON for IMS interprets the screen space entries as if you entered the following commands:

```
DEV 123
DIO
.SGO MONITOR
```

The next examples show you how to conditionally set variables. In the first example, the variable *SYSTEM* sets to *A* if the variable *SWITCH* is previously set to YES; *SYSTEM* sets to *B* if *SWITCH* is previously set to NO; and *SYSTEM* sets to *C* if *SWITCH* is previously set to MAYBE.

```
C.VAR SET &SYSTEM A &SWITCH=YES
C.VAR SET &SYSTEM B &SWITCH=NO
C.VAR SET &SYSTEM C &SWITCH=MAYBE
```

In the following example, the variable TOKEN sets to z/OS only if the product prefix is *OM*.

```
C.VAR SET &TOKEN z/OS PREFIX=OM
```

.VTM

Displays terminal ID and session information for all users that are logged on to OMEGAMON for IMS in VTAM mode.

Type:
Immediate

This command allows a user who is running a multi-session environment that is moderated by KOBVTAM to display information about other KOBVTAM users. You can use this command to monitor and manage access to the VTAM environment. Dedicated mode users do not display.

The following figure is an example of the display.

```
 .VTM  Userid   Terminal  Mode    Session Start        Last Update
+      TSOX07   L65F      VTM   02/11/95 11:45:32   02/12/95 13:14:55
+      TSOX06K  BBLVM06   VTT   02/12/95 13:57:32   IN INITIALIZATION
+      AFOPER   BBLVM04   VTT   02/12/95 11:43:22   02/12/95 13:57:12
+      TSOX21   L616A09   VTS   02/12/95 10:23:31   02/12/95 11:24:31
+      TSOX36A  L674      VTM   02/12/95 13:40:33   02/12/95 13:57:51
+      TSOX04   L655      VTD   02/12/95 11:35:32   02/12/95 12:14:55
```

Figure 17. VTAM display

The **Mode** field indicates the type of session. The options are as follows:

VTD
A director segment that runs under VTAM in a cross memory or cross system mode session.

VTM
A VTAM mode session.

VTS
An ISPF mode session that runs under VTAM.

VTT
A TSO mode session that runs under VTAM.

The Userid field in this display is blank if external security is not used to control logon access.

/WAIT
Controls synchronization of a cross system or cross memory collector with the director.

Type:
INFO-line

Format:
/WAIT ccc

ON
Starts the collector to synchronize with the director.

OFF
Starts the collector not to synchronize with the director.

For information about cross system and cross memory modes, see the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

.WAT
Waits nn seconds before you run the commands that follow.

Type:
Immediate

Format:
.WATnn

The `.WAT` command provides a delay mechanism for the execution of commands that require information from a running command, such as one that involves calculation of a rate.

/XLF OUT
Sends exception logging facility (XLF) data to the printer and resets the log.

Type:
INFO-line

The exception logging facility (XLF) writes exceptions to the XLFLOG. To view or change the defaults for this file, use the OUTP major command and its minors.

For information about using the XLF feature, see Chapter 14, “OMEGAMON for IMS automation and logging features,” on page 317.

The `.XLFOUT` command is the equivalent immediate command.

.XLFOUT
Sends Exception Logging Facility (XLF) data to the printer and resets the log.

Type:
Immediate

The `.XLFOUT` command comments itself out after it runs. See the equivalent INFO-line command, `/XLF OUT`, for more information.

.XLG
Displays and sets printer characteristics for the Exception Logging Facility (XLFLOG).

Type:
Immediate

When you enter the `.XLG` command, a series of keywords displays for setting printer characteristics. If you change more parameters than fits on one line, use the `OUTP` command instead. The keywords for the `.XLG` command (and the `.REP` command) duplicate the minors of the `OUTP` major command.

SYSOUT=
SYSOUT class.

HOLD=
Specifies whether output is to be placed in the hold queue.

COPIES=
Specifies the number of copies to print.

FORMS=
Specifies the form on which to print.

DEST=
Destination, user ID, or both (separated by a period, colon, or slash) to receive report.

FOLD=
Folds lowercase characters to uppercase.

The following keywords also display if their default values are previously modified. If not, you can type in the keyword and define a new value.

ID1=
Requests separator pages and page headers. The argument for ID1 can be:

*****
By default, OMEGAMON for IMS generates separator pages and page headers with the appropriate job name printed in block letters on the pages.

NONE
OMEGAMON for IMS does not generate page headers or separator pages.

cccccccc
OMEGAMON for IMS generates separator pages and page headers with *cccccccc* printed in block letters on the pages. *cccccccc* is up to eight user-defined characters.

ID2=
Defines up to 16 characters on the left of the separator page.

ID3=
Defines up to 16 characters in the center of the separator page.

ID4=
Defines up to 16 characters on the right of the separator page.

DDNAME=
Overrides standard OMEGAMON for IMS ddnames.

If you change any parameter other than FOLD=, OMEGAMON for IMS automatically spins off the XLFLOG file and creates a new one. If SYSOUT is active, then DDNAME is inactive and vice versa. The following table shows the parameters in effect and the default settings for SYSOUT or DDNAME.

Table 5. SYSOUT and DDNAME default values and parameters

Parameters	**Default Values**	
	SYSOUT	**DDNAME**
SYSOUT=	A	(inactive)
HOLD=	NO	(inactive)
COPIES=	1	(inactive)
FORMS=	*NONE*	(inactive)
DEST=	*NONE*	(inactive)
DDNAME=	(inactive)	OIXLFLOG
FOLD=	YES	YES

Table 5. SYSOUT and DDNAME default values and parameters (continued)		
Parameters	**Default Values**	
	SYSOUT	**DDNAME**
ID1=	jobname	jobname
ID2=	(blank)	(blank)
ID3=	(blank)	(blank)
ID4=	(blank)	(blank)

.ZAP

Displays maintenance ZAPs applied.

Type:
Immediate

For information about the **.ZAP**command and its output, see the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

/ZOOM

Starts navigational zoom feature by using the cursor as a pointer.

Type:
INFO-line

The zooming feature is designed to simplify the investigation of system conditions by supplying a detailed level of information at the touch of the Zoom key. /ZOOM substitutes the value above the current cursor position for a variable that is contained in a predefined screen space. The variable substitution allows one zooming screen space to analyze multiple items, such as devices or volume serials. It also facilitates quick investigation of exception conditions.

Note the following points about this feature.

- IBM ships OMEGAMON for IMS with the /ZOOM INFO-line command assigned to PF11 so you can access the zooming screen spaces with a single key. PF11 is referred to as the Zoom key.
- The menu system uses the zooming feature extensively. For example, when you are looking at an exception analysis display, you can place your cursor on an exception name and press PF11. OMEGAMON for IMS zooms to a recommendation screen that gives you suggestions on actions you might want to take.

  In the menu system, when there are fields on a display that respond to the Zoom key, such as exception names or device names, **Zoom PF11** is shown under the INFO-line as shown in the following figure.

```
 ________________ KOISYS   VTM     OI-II    V530./C IA1W 05/25/15 11:31:00    B
> Help PF1       Back PF3        Up PF7        Down PF8        Zoom PF11
```

  Figure 18. Zoom key

  Note: To zoom to exception recommendations from command mode, you can set a command mode PF key to /ZOOM @ZSM.

- You can use zooming in command mode by setting up customized investigative screen spaces and zooming on command or exception names. These screen spaces can contain one or more of the following variables:

  &ZOOM
  Data that is found at the cursor location.

  &ZOOMC
  Command or exception name field (columns 2 - 5).

&ZOOMS
: Originating screen space.

For information about how to create screen spaces, see the *IBM OMEGAMON for IMS on z/OS: OMEGAMON for IMS User's Guide*.

Valid delimiter characters for zoom values (characters that OMEGAMON for IMS recognizes as the beginning or end of the value) are as follows:

: blank

+
: plus sign

( )
: left and right parenthesis

|
: Vertical bar

'
: Single quotation mark

<>
: Greater than and less than sign

=
: Equal sign

*****
: Asterisk

Chapter 3. The user profile facility

OMEGAMON for IMS profiles control the characteristics of an active OMEGAMON for IMS session. You can configure three types of profiles: the IBM-provided profile, the installation profile, and individual user profiles.

Installation profiles are intended to be created by the installer or other authorized personnel, and are typically specified as the default for the site or for a system. You can configure as many individual user profiles as you require. You can create the profile, save it by assigning it a unique two-character identifier, and specify the required profile ID at logon.

Installation profiles and individual user profiles are created by customizing the settings of the profile definition commands and issuing a profile save command. The save command for an installation profile is IPRF; for a user profile, it is PPRF. You can change the settings of user profile options at any time during a session, and the new settings take effect immediately.

The following table summarizes the profile definition commands. For more information about setting profile options, see *IBM OMEGAMON for IMS on z/OS: OMEGAMON for IMS User's Guide*.

Table 6. (1 of 2) Profile definition commands

Command Function	**Command**	**Description**
Controlling session and display options	OPTN	Activates and deactivates the following components: • Automatic Screen Facility (ASF) • Timed Screen Facility (TSF) • Exception Logging Facility (XLF) • terminal bell • log Controls these display characteristics: • the date (US or European) • minor commands (upper or lowercase) • all command output (upper or mixed case) • scroll amount (page or cursor) • the first screen when OMEGAMON for IMS is started • the interval for the terminal bell **Note:** The setting for the ZEROS keyword is not saved in a profile.

Table 6. (1 of 2) Profile definition commands (continued)		
Command Function	**Command**	**Description**
Setting print output options	OUTP	The minor command settings that are saved in a profile are as follows: COPY specifies number of copies to print. DDNM specifies ddname to override standard ddname. DEST specifies the report destination. FOLD specifies whether lowercase is folded to uppercase. HOLD specifies that the output is placed in the hold queue and retrieved from TSO. LNCT sets number of lines per page on report. SOUT defines the SYSOUT class of the report. **Note:** Other minors of OUTP are not saved in profiles.
Setting color options	.SCC	You can set color, highlighting, and extended attribute options for each field on the OMEGAMON for IMS display.
Setting operational parameters	.SET	Sets parameters for: • the screen space fetch feature (.FGO) • the interval for OMEGAMON for IMS cycles • the number of entries in the device name table • guarding against loops that are caused by the PEEK command • the automatic updating delay cycle • the size of the REPORT file for logging screens • the size of the work area for the PEEK command • The Workload Profile Facility status mode
Setting installation profile options	IOPT	Determines whether OMEGAMON for IMS storage is page-fixed and whether OMEGAMON for IMS issues a DASD RESERVE when users save a screen space member.
Defining Timed Screen Facility entries	.TSF	See Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317.
Setting exception analysis options		See table in Chapter 4, "Customizing and creating exceptions," on page 61.
Customizing IMS messages	MSGD	Assign names to IMS messages for use in exception analysis. See the full description of this command in Chapter 4, "Customizing and creating exceptions," on page 61.

Multiline input

Some profile definition commands use a multiline input facility, for example, the OPTN, .SCC, and .SET commands. When you enter these commands, OMEGAMON for IMS displays the current settings for the options in a multiline table.

The following considerations apply to multiline tables:

- You can modify any display line that begins with a colon in column 1.

- Blanking out fields has no effect; OMEGAMON for IMS redisplays the line on the next cycle.
- To change a setting, type over the displayed value and press ENTER.

Profile maintenance commands

Use the IPRF command to configure the installation profile. Use the PPRF command for user profiles.

IPRF
Saves or deletes the installation-defined default profile.

Type:
Immediate

Format:
IPRF SAVE|DELETE

SAVE
Save the installation profile in the data set defined by the RKOIPFSV DD statement.

DELETE
Deletes the installation profile from the data set defined by the RKOIPFSV DD statement.

OMEGAMON for IMS automatically assigns the two-character code /I to the installation profile. /I is used on the USER= startup parameter to load this profile, and it displays on the INFO-line during the session.

The installation-defined profile for OMEGAMON for IMS for IMS is stored as member name OIINSTAL.

PPRF
Saves, deletes, comments, or lists the user's session profile.

Type:
Immediate

Format:
PPRF SAVE|DELETE|COMMENT|LIST cc

SAVE
Saves the user profile in the data set defined by the RKOIPFSV DD statement.

DELETE
Deletes the user profile from the data set defined by the RKOIPFSV DD statement.

COMMENT
Displays a description of the current profile; you can change it dynamically.

LIST
Lists all members of the user profile data set.

cc
Specifies the two-character user profile identifier. To start subsequent sessions with this profile, specify this value in the USER= startup parameter.

If you do not specify *cc*, the PPRF command uses the current value of the USER= startup parameter.

Controlling session and display options

Use the OPTN command parameters to set session and display options.

OPTN
Sets session control and display options.

Type:
Immediate

OPTN displays its current settings in the following format:

```
OPTN
:     ASF          = OFF          BELL         = OFF
:     BELLINT      =    60.00     DATEFORMAT   = USA
:     FIRSTSCREEN  = KOINITZZ     LOG          = OFF
:     MINORCASE    = LOWER        SCREENCASE   = MIX
:     SCROLL       = PAGE         TSF          = OFF
:     XLF          = OFF          ZEROS        = OFF
```

ASF
Turns the Automatic Screen Facility (ASF) ON or OFF. See Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317 for information about this facility.

BELL
Turns the audible alarm ON or OFF.

BELLINT
Sets the minimum interval (in seconds) between rings of the bell. A valid value is an integer between 5.00 and 99.00.

DATEFORMAT
Sets display format for the date (*mm/dd/yy* or *dd/mm/yy*). Valid values are USA or EUROPEAN.

FIRSTSCREEN
Identifies the name of the first screen space to run, or it might FGO to another screen space.

LOG
Turns the log ON or OFF.

MINORCASE
Sets the display case for minor commands. Valid values are UPPER and LOWER.

SCREENCASE
Sets the display case for screen output. Valid values are UPPER and MIX.

SCROLL
Sets the default scroll amount. Valid values are PAGE, which scrolls an entire screen at a time, and CSR, which scrolls from the cursor position.

TSF
Turns the Timed Screen Facility (TSF) ON or OFF. For information about this facility, see Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317.

XLF
Turns the Exception Logging Facility (XLF) ON or OFF. For information about this facility, see Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317.

ZEROS
Sets how zeros display on your terminal. When this parameter is ON, OMEGAMON for IMS displays the numeral 0 in fields that have a value of zero. When it is OFF, OMEGAMON for IMS displays a blank in these fields. This value is not saved in a profile.

Setting print output options

The **OUTP** major command and its minor commands control the printing of XLFLOG and REPORT files.

OUTP
Controls the characteristics of log files.

Type:
Major

Format:
OUTP cccccc

The variable *cccccc* is either XLFLOG or REPORT.

The OUTP major command displays column headings for pending and current values that are associated with all of its minor commands follows.

```
 OUTP REPORT
+                            |----- Pending ------|----- Current ------|
```

When you type an OUTP minor command followed by the new value you want to assign and press ENTER, the new value displays in the pending value column. It becomes the active value when you reallocate the log with /LOGOUT, .LOGOUT, /XLFLOG, or .XLFLOG.

- The log routes to one of the following data set types:
- SYSOUT, the initial data set type for OUTP. This file is designated FREE=CLOSE, which means that every time you enter the /LOGOUT, .LOGOUT, /XLFOUT, or .XLFOUT command, the REPORT log or the XLFLOG automatically spins off and is available for printing.
- A sequential data set, with a ddname you specify. Its DCB attributes are LRECL=nn, where *nn* is the screen column width plus 1; RECFM=FBA; and BLKSIZE is a multiple of LRECL. The DISP parameter can be set to MOD, SHR, or OLD.

CAUTION: Logging to a data set is optional and is *not recommended*. If you use DISP=SHR or DISP=OLD, any action to close and reopen the log file reinitializes the data set and deletes any existing log information.

Only one session at a time per address space must use a specific ddname. Otherwise, you get interleaved output.

Note: Some prior releases of OMEGAMON for IMS used ddname OIREPORT if it is present. This release also looks for ddname OIREPORT if present, but accesses it only if it is not currently in use.

Each OUTP minor command controls one XLFLOG or REPORT file characteristic. That characteristic is under the control of ddname or SYSOUT. When SYSOUT is active, ddname is inactive, and vice versa. Parentheses around a value in the OUTP display indicate that it is inactive. The following table shows the minors and the initial settings for the SYSOUT or ddname data sets.

Table 7. OUTP minor commands with initial settings

Parameters	**Initial Values**	
	SYSOUT	DDNAME
SOUT	A	(inactive)
HOLD	NO	(inactive)
COPY	1	(inactive)
FORM	*NONE*	(inactive)
DEST	*NONE*	(inactive)
DSTU	userID	(inactive)
DDNM	(inactive)	*DYNAMIC
FOLD	YES	YES
LNCT	60	60
ID1	jobname	jobname

Table 7. OUTP minor commands with initial settings (continued)		
Parameters	**Initial Values**	
ID2	(blank)	(blank)
ID3	(blank)	(blank)
ID4	(blank)	(blank)

COPY

Specifies the number of copies to print.

The COPY minor is under SYSOUT control.

Type:
Minor of OUTP

Format:
COPY nn

If you set the HOLD minor command to YES, OMEGAMON for IMS ignores the COPY command. If the value you type is not valid, OMEGAMON for IMS redisplays it where you typed it and does not transfer it to the pending column.

DDNM

Directs output to a ddname.

The DDNM minor removes control from SYSOUT.

Type:
Minor of OUTP

Format:
DDNM cccccccc

If the value you type is not valid, OMEGAMON for IMS redisplays it where you typed it and does not transfer it to the pending column.

DEST

Specifies the destination to receive the output.

The DEST minor is under SYSOUT control.

Type:
Minor of OUTP

Format:
DEST cccccccc

The default is ***NONE***, which sends output to the local printer. The destination can be a terminal, a node, a remote work station, a local device, or group of devices, or a user ID.

Your site determines valid destinations. OMEGAMON for IMS checks your site's table for valid destinations only when you reset the log (with .LOGOUT or .XLFOUT). During initialization, OMEGAMON for IMS checks only syntax validity.

If the destination is a specific user ID (at the device destination), use the DSTU minor command for the user ID.

DSTU

Specifies the destination user ID to receive a report.

The DSTU minor is under SYSOUT control.

Type:
Minor of OUTP

Format:
DSTU cccccccc

The default is ***NONE***, which sends output to the local printer. Enter the destination user ID in the format that is established for your site.

Your site determines valid destinations. OMEGAMON for IMS checks your site's JES parameters for valid destinations only when you reset the log (with .LOGOUT or .XLFOUT). During initialization, OMEGAMON for IMS checks only syntax validity.

FOLD
Changes lowercase characters to uppercase.

FOLD is active under ddname or SYSOUT.

Type:
Minor of OUTP

Format:
FOLD ccc

YES
Lowercase characters are changed to uppercase before printing (default).

NO
Lowercase characters are not changed to uppercase before printing.

If the value you type is not valid, OMEGAMON for IMS redisplays it where you typed it and does not transfer it to the pending column.

FORM
Specifies the name of the form on which to print.

The FORM minor is under SYSOUT control.

Type:
Minor of OUTP

Format:
FORM ccc

The default value is ***NONE***, which means that OMEGAMON for IMS uses the form that is defined as the default for the printer at that destination.

If the value you type is not valid, OMEGAMON for IMS redisplays it where you typed it and does not transfer it to the pending column.

If you change ***NONE*** to a value (such as HOLE), and you then want to return to ***NONE***, type an asterisk(*).

HOLD
Specifies that output is placed in the hold queue.

The HOLD minor is under SYSOUT control.

Type:
Minor of OUTP

Format:
HOLD ccc

YES
Places the print file on the hold queue, and allows it to be retrieved from TSO.

NO
Does not place the print file on the hold queue (default).

If the value you type is not valid, OMEGAMON for IMS redisplays it where you typed it and does not transfer it to the pending column.

ID1

Requests separator pages and page headers that identify output from different OMEGAMON for IMS sessions.

ID1 is active under ddname or SYSOUT.

Type:
Minor of OUTP

Format:
ID1 cccccccc

*
By default, OMEGAMON for IMS generates separator pages and page headers with the appropriate job name printed in block letters on the pages.

NONE
OMEGAMON for IMS does not generate separator pages and page headers.

cccccccc
OMEGAMON for IMS generates separator pages and page headers with *cccccccc* printed in block letters on the pages. *cccccccc* is up to eight user-defined characters.

ID2

Defines up to 16 characters at the left of the separator page.

ID2 is active under ddname or SYSOUT.

Type:
Minor of OUTP

Format:
ID2 ccc...cc

When separator pages and page headers are requested with ID1, ID2 can define up to 16 characters to display justified below the block letters on the left of the separator page.

ID3

Defines up to 16 characters in the center of the separator page.

ID3 is active under ddname or SYSOUT.

Type:
Minor of OUTP

Format:
ID3 ccc...ccc

When separator pages and page headers are requested with ID1, ID3 can define up to 16 characters to display centered below the block letters on the separator page.

ID4

Defines up to 16 characters at the right of the separator page.

ID4 is active under ddname or SYSOUT.

Type:
Minor of OUTP

Format:
ID4 ccc...ccc

When separator pages and page headers are requested with ID1, ID4 can define up to 16 characters to display justified below the block letters on the right of the separator page.

LNCT

Sets the number of lines per page for the REPORT or XLFLOG file output.

Type:
Minor of OUTP

Format:
LNCT nn

SOUT
Removes control from a sequential data set, and directs it to the specified SYSOUT class.

The SOUT minor removes control from the ddname that is used for your XLFLOG or REPORT log and directs it to SYSOUT.

Type:
Minor of OUTP

Format:
SOUT c

The variable *c* represents the SYSOUT class.

Setting color options

Use the .SCC command to control color and highlighting options in OMEGAMON for IMS displays.

.SCC
Sets display color or highlighting for text and commands by type.

Type:
Immediate

The .SCC command displays and sets options that determine how highlighting and color is used when an OMEGAMON for IMS screen is sent to a user's terminal. Because display characteristics depend on the type of terminal you are supporting, the .SCC command contains keywords that, in combination, accommodates any of the various 3270-type devices. The following figure shows the typical screen display format:

```
 .SCC
: Display=ccccc                              ProfileDefinitionMode=OFF
: ExtendedHighlighting=ccc                   DBCS=OFF
+
: Major=ccccccccc  Minor=cccccccc   Immed=cccccccc   Default=cccccccc
+
+ XACB Display Options:
: Clr1=cccccccc    Clr2=cccccccc    Clr3=cccccccc    Clr4=cccccccc
: Clr5=cccccccc    Clr6=cccccccc    Clr7=cccccccc
```

Figure 19. .SCC screen display format

Device support

Color and highlighting capabilities in display output depends on whether the device supports the capabilities.

Non-EDS
Devices that do not support an extended data stream (EDS), but support highlighting. There are two types of non-EDS devices:

- Monochrome non-EDS devices, which use high or low intensity to distinguish fields in a display.
- Color non-EDS devices, which use different colors rather than high and low intensity to distinguish fields in a display. The colors that display depend on whether the device is a two-base color or four-base color device, and whether a field is protected or unprotected. For example, most four-base color devices display high intensity fields in red (unprotected) and white (protected), and low intensity fields in green (unprotected) and blue (protected).

Monochrome EDS
Monochrome devices that support the extended data stream. A field might be displayed in either high or low intensity in combination with the extended highlighting attributes (flashing, reverse video, and underscoring).

Color EDS
: Color devices that support the extended data stream. A field might be displayed in any of the seven extended colors in combination with the extended highlighting attributes.

.SCC command keywords and values

You can set color and highlighting options in the .SCC multiline display. Or you can type a keyword and value on the .SCC command line.

For any keyword or value, you need only type as many letters as it takes to make an entry unique. When you type a keyword and value on the command line after .SCC, the command comments itself out and flags itself DONE at the end of the line. This facility allows for changes to .SCC values from within screen spaces.

Profile definition mode

Before customizing your color and highlighting definitions, determine which mode of the `.SCC` command is appropriate. The mode is controlled by the **`ProfileDefinitionMode`** keyword, as shown in the following figure.

```
 .SCC
: Display=ccccc                               ProfileDefinitionMode=ON/OFF
: ExtendedHighlighting=ccc                    DBCS=OFF
```

Figure 20. Profile definition mode

OFF

Changes to the `.SCC` command affects only the current OMEGAMON for IMS session.

ON

Intended for use when you create or change a user profile without altering the current session. To change your settings without altering your current session, turn on Profile Definition mode, and alter the settings. After you define the settings, issue a profile save command. Then, change the settings back, and turn off Profile Definition mode.

The new settings might also take effect during the current session if you then change the value of the **`Display=`** keyword. Set this keyword to ON to configure options for different types of terminals on the same screen.

Display intensity or color

The **`Display`** keyword has four possible settings.

*Figure 21. **`Display`** keyword*

```
 .SCC
: Display=BASIC|HIGH|LOW|COLOR
```

BASIC

Set HIGH or LOW intensity for fields on monochrome terminals or non-EDS color terminals.

Note: When the BASIC display option is used on color devices that *can* support EDS, OMEGAMON for IMS treats that device as a 4-base color terminal (non-EDS).

HIGH

Specifies that all fields be displayed in high intensity.

LOW

Specifies that all fields be displayed in low intensity.

COLOR

For color EDS terminals only, you can specify the color of each field.

Note: In cases where OMEGAMON for IMS is given a color value instead of an intensity value or vice versa, it makes the following internal conversion:

- On a non-EDS terminal, values of Green and Blue translate to low intensity; all other color values translate to high intensity.
- On an EDS terminal, a value of HI translates to the color Red; a value of LO translates to the color Green.

Extended highlighting

The **ExtendedHighlighting** keyword is used only with devices that support the extended data stream.

```
: ExtendedHighlighting=ON|OFF
```

OFF
Extended highlighting features are not available. Use with non-EDS devices.

ON
Extended highlighting features are available. When **Display=COLOR**, this value is automatically set to ON. Extended attributes are not supported in ISPF mode.

CAUTION: Do not set **ExtendedHighlighting=ON** unless you have a terminal that supports an extended data stream (or unless you have **ProfileDefinitionMode=ON**). If you do this accidentally, you might get a PROGnnn or a screen erasure error. You can press the ATTN or PA1 key to resume the session, but this action also clears the current security authorization and the current screen space.

Display fields

The following figure shows the display format of .SCC with **ProfileDefinitionMode=ON**. It also shows the display field keywords and valid values.

For the **Display=COLOR** option, color names can be Red, Green, White, Blue, Pink, Yellow, or Turquoise.

For the **Display=BASIC** option, highlighting values can be HIGH or LOW.

The value of the **Default** keyword can be used as a variable definition for the Major, Minor, Immed, and XACB Display Options. In Figure 22 on page 57, the default value is abbreviated as DEF.

```
 .SCC
: Display=COLOR|BASIC                        ProfileDefinitionMode=ON
: ExtendedHighlighting=ON
+
: Display=COLOR Options:
:  MajorC=color|DEF  MinorC=color|DEF  ImmedC=color|DEF  DefaultC=color
+
+  XACB Display Options:
:  Clr1C=color|DEF   Clr2C=color|DEF   Clr3C=color|DEF   Clr4C=color|DEF
:  Clr5C=color|DEF   Clr6C=color|DEF   Clr7C=color|DEF
+
: Display=BASIC Options:
: MajorB=HI|LO|DEF  MinorB=HI|LO|DEF  ImmedB=HI|LO|DEF  DefaultB=HI|LO
+
+  XACB Display Options:
:  Clr1B=HI|LO|DEF   Clr2B=HI|LO|DEF   Clr3B=HI|LO|DEF   Clr4B=HI|LO|DEF
:  Clr5B=HI|LO|DEF   Clr6B=HI|LO|DEF   Clr7B=HI|LO|DEF
```

Figure 22. Display format of .SCC

The field names that you can control with .SCC are:

Major
Controls color or highlighting for major commands and their output.

Minor
Controls color or highlighting for minor commands and their output.

Immed
Controls color or highlighting for immediate commands and their output.

Default
Controls color or highlighting for other unprotected fields (for example, error message text, help text).

XACB Options
Controls exception analysis message text. The keywords Clr1 through Clr7 can be used as substitutes for the color names (for example, Red or Blue) or highlighting (HI or LO) when customizing exception messages with the XACB command.

Color and highlighting for your terminal type

The following table shows the possible color and highlighting setting variations according to the type of terminal you have.

Table 8. Color/Highlighting settings in .SCC

If you have...	and you want...	keywords and possible settings are...
A non-EDS terminal	some fields in high intensity and some fields in low intensity	Display=BASIC ExtendedHighlighting=OFF Major, Minor, Immed, XACB options=HI|LO|DEF Default=HI|LO
	all fields in high intensity	Display=HI ExtendedHighlighting=OFF (All other settings default to HI.)
	all fields in low intensity	Display=LO ExtendedHighlighting=OFF (All other settings default to LO.)
A monochrome EDS terminal	some fields in high intensity and some fields in low intensity, plus blinking, underscoring, or reverse video	Display=BASIC ExtendedHighlighting=ON Major, Minor, Immed, XACB options=HI|LO|DEF Default=HI|LO
	all fields in high intensity, plus blinking, underscoring, or reverse video	Display=HI ExtendedHighlighting=ON (All other settings default to HI.)
	all fields in low intensity plus blinking, underscoring, or reverse video	Display=LO ExtendedHighlighting=ON (All other settings default to LO.)
A color EDS terminal	to specify the color of each field, plus blinking, underscoring, or reverse video	Display=COLOR ExtendedHighlighting=ON Major, Minor, Immed, XACB options=color|DEF Default=color

Setting operational parameters

Use the .SET immediate command to display data about the OMEGAMON for IMS environment in table form. You can change data on any of the output lines.

.SET

Sets operation control parameters.

Type:

Immediate

Format:

```
.SET
:     FGOLIMIT     =       64     FGOLOOP      = OFF
:     GDEVUCBS     =      200     INTERVAL     =    10.00
:     IODELAY      =        5     LOOPCOUNT    =    15000
:     LOOPTIME     =     5.00     PAGELIMIT    =      400
:     PEEKSIZE     =     4096     STATUSMODE   = OFF
```

Figure 23. .SET command display

To change a setting, move the cursor to the value you want to change, type the new value over the current value, and press Enter.

FGOLIMIT

Specifies the maximum number of consecutive .FGO screens that can run before OMEGAMON for IMS detects a loop and sets **.FGOLOOP=ON**. The maximum number is 1000.

FGOLOOP

Turns .FGO screen loop detection ON or OFF. If OMEGAMON for IMS detects an .FGO loop, it sets this keyword to ON and then treats subsequent **.FGO** commands as **.SGO** commands (the **.FGO** command runs a screen space without displaying it; the **.SGO** command displays each screen space that it runs.) You can also set **.FGOLOOP=ON** if you want to test screen spaces that you link together with **.FGO** commands.

GDEVUCBS

Specifies the number of entries in the device name table for the GDEV command. The maximum number is 4000.

INTERVAL

Specifies the interval in seconds between automatic updates. This interval is called an OMEGAMON for IMS cycle. The following restrictions apply:

- This value is effective only in dedicated mode or VTAM mode. You can, however, be operating in any mode when you define a new value for saving it in a user profile.
- The maximum interval is 99.00 seconds.
- VTAM mode does not allow an interval shorter than five seconds.
- Dedicated mode does not allow an interval shorter than 0.5 seconds.

IODELAY

In automatic updating, the number of cycles to delay the next screen refresh after you move the cursor. The maximum number is 100.

LOOPCOUNT

The maximum number of control blocks that the PEEK command can test before OMEGAMON for IMS detects a loop. The valid range is 1–1000000.

LOOPTIME

The threshold (in seconds) for OMEGAMON for IMS built-in loop detection. The default is 150 seconds. The maximum value is 300.

PAGELIMIT

The size (in pages) of the REPORT file that is used to log OMEGAMON for IMS screens. The maximum is 99999. This number dynamically decreases as the log is printing to reflect the number of pages that remain before the limit is reached. Be sure to check this parameter (and reset it, if necessary) before you save a profile.

PEEKSIZE
The size (in bytes) of the PEEK buffer. The maximum is 33553408.

STATUSMODE
This parameter does not apply to OMEGAMON for CICS.

Setting installation performance options

Use the IOPT command to set performance options.

IOPT
Assigns global OMEGAMON for IMS performance options.

Type:
Immediate

Because this command determines how OMEGAMON for IMS runs for the entire site, you might want to use this command at installation time and then restrict access to the command.

```
 IOPT
:     NONSWAP      = OFF          PAGEFIX      = OFF
:     RESERVE      = OFF          TSOPFIX      = OFF
```

Figure 24. IOPT command

NONSWAP
This parameter does not apply to OMEGAMON for IMS.

PAGEFIX
Indicates whether OMEGAMON for IMS storage is page-fixed. Possible values for this are ON and OFF; the default is OFF. If you change the value to ON, your change does not take effect until the next session.

This option requires that OMEGAMON for IMS is APF-authorized.

RESERVE
Indicates whether OMEGAMON for IMS issues a DASD RESERVE when users save a member in the RKOIPFSV data set. Possible values are ON and OFF; the default value is OFF.

TSOPFIX
This parameter does not apply to OMEGAMON for IMS.

You can enter the IOPT command followed by the keyword and setting you want on the same line.

Chapter 4. Customizing and creating exceptions

To better monitor your system, you can customize exceptions. You can also create your own exceptions for your user profiles, which display in the menu and command interface.

Controlling exception analysis

You can use the exception analysis immediate commands to control exception analysis and display status.

The following table summarizes the exception analysis control commands:

Table 9. Exception analysis control commands

Command Function	Command	Description
Starting exception analysis	XIMS	Starts exception analysis for all exceptions.
	XGRP	Starts exception analysis for the specified exception group.
Summarizing exception activity	XSUM	Displays a summary of the last and worst trip values for all exceptions.
	XTRP	Displays a summary of the last and worst trip values for tripped exceptions.
Defining exception characteristics	GDFN	Displays existing groups (both IBM-defined and user-defined). You can also define new groups.
	LEXC	You can specify the order for OMEGAMON for IMS to sample exceptions and display messages.
	MSGD	Define IMS messages for exception analysis.
	XACB	For each individual exception, you can define parameters for XLF and ASF. • Set a threshold. • Control the state (For example, ON, OFF). • Define display characteristics for warnings. • Request audible alarm when the exception trips. • Control the frequency for OMEGAMON for IMS sampling.
	XTXT	You can define the message to display when no exceptions trip.
Controlling Groups	XGSW	You can control the state (for example, ON, OFF) of exceptions by group.

Starting exception analysis

Use the XGRP and XIMS commands to start exception analysis for selected groups only or for all groups.

XGRP

Starts exception analysis for exception group *cc*.

The XGRP command starts only those exceptions that belong to the specified group.

Type:

Immediate

Format:

XGRPcc

The variable *cc* is the ID for a group that is defined with the GDFN command or for an IBM-provided default group. Use GDFN to display currently defined groups.

AE

Application execution exceptions

AL

Pool usage exceptions

DB

Database exceptions

DL

DASD logging exceptions

FA

Fast path exceptions

FR

Largest free block (fragmentation exceptions)

IM

IMS internal exceptions

IV

IMS virtual storage exceptions

OS

z/OS resource exceptions

ST

Static exceptions

VS

VSAM exceptions

XR

XRF exceptions

For example, to start only the exceptions that belong to the IMS internal group (IM), enter XGRPIM as shown in Figure 25 on page 62:

```
 XGRPIM   OMEGAMON for IMS Group Exception Analysis
+      Message Dequeue rate = .06/second  (Low)
+      Output Queue length for logical terminal ‘MTO’ = 5
+      Output Queue length for logical terminal ‘MTOPRINT’ = 5
```

Figure 25. XGRPIM display

XIMS

Starts exception analysis for all groups.

Type:

Immediate

Format:
cXIMS

Use the label *L* to display the four-character exception names that are started by XIMS.

The XIMS command can start over 150 different exception conditions. Exception analysis groups the exceptions into exception groups. See the **XGRP** command for a list of exception groups.

You can turn on or off each exception individually. Some exceptions require that you set a threshold. If a 3279 color display terminal is available, you can assign each exception message a specific color, which causes more important messages to stand out. If the OMEGAMON for IMS terminal supports the audible alarm (or bell), you can request that the alarm ring when certain exceptions occur. You can set the options for all of these exceptions with the **XACB** command and save your settings in a profile with the PPRF command. See "Profile maintenance commands" on page 49 for an explanation of PPRF.

The static exceptions (type ST) select conditions that do not change (such as no $$IMSDIR table defined).

By default, the **XIMS** command does not display a message for a static exception that persists for more than five cycles. However, you can use the **XGRP** command to display static exceptions at any time. If you want to change the setting for an exception in the static group so that it displays after five cycles, use the Stop= parameter of the **XACB** command.

Summarizing exception activity

Use the XSUM and XTRP commands to display a summary of exception activity.

XSUM
Displays a summary of exceptions and their status.

Type:
Immediate

Format:
XSUM GROUP=cc LIST={A|I} RESET

(blank)
By default, XSUM with no keywords displays all of the exception groups in alphabetical order.

GROUP
The group ID (*cc*) can be any of the same exception groups that are used with the XGRP command. Enter a two-character group ID to summarize the exceptions of one group only.

LIST
The value can be *A* or *I*.

A
By default, lists exceptions in alphabetical order.

I
Lists exceptions in the order in which they run, as specified by the **LEXC** command.

RESET
Resets the last and worst values back to zero. Does not reset the cumulative value.

Figure 26 on page 64 shows an example of a partial XSUM display.

```
 XSUM
+----------+----------+-----------+--------------+------------+---------------+
+ DNRS      Threshold  Trip Value  Time Occurred  Total Trips  Trips Since Reset
+ State=On
+ Last                             06/18  17:07:46         14                 14
+ Worst                            06/18  17:07:46
+ Group=OS Limit=3  Persist=2  Auto=OFF  Log=NO
+----------+----------+-----------+--------------+------------+---------------+
+ DRDY      Threshold  Trip Value  Time Occurred  Total Trips  Trips Since Reset
+ State=On
+ Last                             NOT TRIPPED  0                  0
+ Worst
+ Group=OS  Limit=3  Persist=2  Auto=OFF  Log=NO
+----------+----------+-----------+--------------+------------+---------------+
+ WSHI      Threshold  Trip Value  Time Occurred  Total Trips  Trips Since Reset
+ State=Test     2500
+ Last                      3640K  06/18  16:09:30          6                  8
+ Worst                     3650K  06/18  16:07:46
+ Group=OS  Limit=3  Persist=2  Auto=OFF  Log=NO
+----------+----------+-----------+--------------+------------+---------------+
+ WSLO      Threshold  Trip Value  Time Occurred  Total Trips  Trips Since Reset
+ State=Test      300
+ Last                       270K  06/18  16:09:30          5                  9
+ Worst                      265K  06/18  16:07:46
+ Group=OS  Limit=3  Persist=2  Auto=OFF  Log=NO
+----------+----------+-----------+--------------+------------+---------------+
```

Figure 26. XSUM display

The XSUM command displays the current settings for the exception state (**Staten/OFF/TEST**), the group to which it is assigned (**Group=**), and XLF or ASF settings (**Limit=**, **Persist=**, **Auto=**, and **Log=**). In addition, it displays the last and worst values for the following fields.

Threshold
The current threshold value that is set for this exception.

Trip Value
The value that caused this exception to trip.

Time Occurred
The date and time the exception last exceeded its threshold.

Total Trips
The number of times this exception exceeded its threshold during the current session.

Trips Since Reset
The number of times this exception exceeded its threshold since the last and worst values were reset.

XTRP
Displays a summary of tripped exceptions for a group.

Type:
Immediate

Format:
XTRP GROUP=cc LIST={A|I} RESET

(blank)
By default, the **XTRP** command with no keywords displays all of the exception groups in alphabetical order.

GROUP
The group ID (*cc*) can be any of the same exception groups that are used with the XGRP command. Enter a two-character group ID to display the exceptions of one group only.

LIST
The value can be A or I.

A
By default, lists exceptions in alphabetical order.

I

Lists exceptions in the order in which they run, as specified by the **LEXC** command.

RESET

Resets the last and worst values back to zero. Does not reset the cumulative value.

The XTRP display is the same as the XSUM display, but shows only tripped exceptions rather than all exceptions.

Defining exception characteristics

Use the exception characteristics definition commands to customize your exceptions. Then, you can save the new definitions in a user profile.

GDFN

Defines or lists exception groups for exception analysis.

With this command, exceptions can be organized by groups such as hardware, software, system services, critical applications, tape and disk drives, and online applications. Then, when you invoke exception analysis by group with the **XGRP** command, critical and related exceptions display together on the display.

Type:

Immediate

Format:

GDFN GROUP=cc NAME='cc...cc' LIST=cccc DELETE={GROUP|EXCEPTION} POSITION=nn

(blank)

Lists user-defined and IBM-defined exception groups, and the exceptions included in each group.

GROUP

Specifies the two-character exception group ID. To list the entries for an existing group, enter this keyword and the group ID.

NAME

Specifies a 25-character user-defined description of the exception group. Enclose in single quotes if there are blanks or special characters in the name.

LIST

Adds exceptions to the exception group specified with the `GROUP=` keyword. An exception can associated with only one group at a time.

DELETE

The value can be `GROUP` or `EXCEPTION`.

GROUP

Deletes the exception group specified by the `GROUP=` keyword.

EXCEPTION

Deletes the exceptions specified with `LIST=` from the group specified by the `GROUP=` keyword.

POSITION

Specifies the order in which GDFN displays defined groups. The variable *nn* is a position number for the specified group relative to the other groups.

For example, to define the group SP and its related exceptions, you can enter the following.

```
GDFN GROUP=SP POSITION=1 NAME='SYSTEMS PROGRAMMER'
GDFN GROUP=SP LIST=DISP,ROLO,LDMB,PBTR,DNRS,TNRS,DRDY,TRDY
GDFN GROUP=SP LIST=WSHI,IMHI,IMLO,PIMC,ACWA,ACEA
```

To delete specific exceptions from group TX, specify DELETE=EXCEPTION and the list of exceptions as follows:

```
GDFN GROUP=TX DELETE=EXCEPTION LIST=WSHI,WSLO
```

Use the DELETE=GROUP option to delete the group TX and all its related exceptions.

```
GDFN GROUP=TX DELETE=GROUP
```

LEXC
Lists and sets the order of exceptions for exception analysis sampling.

Type:
Immediate

The LEXC command displays the order in which OMEGAMON for IMS executes exceptions. To change the sequence of exception messages displayed by the XIMS command, you can dynamically reorder the execution sequence.

The following figure shows a partial LEXC display. To change the order of the exceptions, type over an exception name or its number.

```
 LEXC
:      INAC =  1   DISP =  2   DNRS =  3   TNRS =  4   DRDY =  5   TRDY =  6
:      IORC =  7   SPAH =  8   QBKH =  9   SMGH = 10   LMGH = 11   RDSH = 12
:      DBLH = 13   MFSH = 14   TMFH = 15   ACBH = 16   OSBL = 17   SAPW = 18
:      ITWH = 19   SDSP = 20   CROL = 21   ARSP = 22   CSVC = 23   PSVC = 24
            .           .           .           .           .           .
            .           .           .           .           .           .
            .           .           .           .           .           .
```

Figure 27. LEXC display

You can also type in exceptions with new order numbers on the command line following the LEXC command, as shown in the following example.

```
LEXC DNRS=1 WSHI=2 DRDY=3
```

MSGD
Defines messages for message exception analysis.

Type:
Immediate

Format:
MSGD [ADD cccccccc|DELETE cccccccc|LIST cccccccc] [OPTION={BUFSHOW|BUFFER|NOSHOW}]

(blank)
Lists all defined messages.

ADD
Adds a specified IMS message.

DELETE
Deletes a specified message.

LIST
Lists a specified message or series of messages.

cccccccc
Specifies the IMS message ID. These characters match the first characters of the IMS message text. There are no assumed blanks appended to any message ID of less than eight characters. You can put the message ID in quotes to add blanks to the message text.

You can use an asterisk (*) as a wildcard character. There is no need to use the asterisk (*) to delineate the end of a message ID string, because it is assumed with an unquoted string.

```
MSGD ADD DFS206
MSGD ADD DFS206*
```

Both examples translate to the same compare string.

```
MSGD ADD DFS206
MSGD ADD 'DFS206 '
```

In the previous example, the first entry matches the IMS message ID DFS206 and DFS2061I. The second entry matches only the IMS message ID DFS206.

OPTION
The OPTION parameter applies only when the ADD keyword is specified. Valid options are BUFSHOW, BUFFER, and NOSHOW.

BUFSHOW
Displays the message through exception analysis and retains it for display by the PMSG immediate command. BUFSHOW is the default.

BUFFER
Retains the message for display by the PMSG immediate command.

NOSHOW
Does not display the message in exception analysis or with the PMSG command. However, if the exception occurs, you can set it to start the exception logging facility (XLF). See Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317 for an explanation of how to log exceptions.

Note: You must issue the ICNS command to activate message exception analysis before you can define messages with the MSGD command. See "IMS messages and IMS-related z/OS messages (ICNS)" on page 287 for more information about the ICNS command.

The MSGD command assigns a number (Mnnn) that becomes an exception name. You can define up to 200 messages. Figure 28 on page 67 shows output from the MSGD command when it is entered without any keywords.

```
 MSGD
+ M002 DFS428    OPTION=BUFSHOW
+ M003 DFS730I   OPTION=BUFSHOW
+ M004 DFS236    OPTION=BUFSHOW
+ M005 DFS286I   OPTION=BUFFER
+ M006 DFS551I   OPTION=BUFSHOW
```

Figure 28. MSGD command output

You can also add z/OS messages that are generated by an IMS region, as illustrated in the following figure.

```
 MSGD ADD ICE074 OPTION=BUFSHOW
+ M007 ICE074   OPTION=BUFSHOW
MESSAGE HAS BEEN ADDED
```

Figure 29. z/OS messages

Note: Messages defined with the MSGD command exist as exceptions during the current OMEGAMON for IMS session only, unless you save them in a profile with the PPRF command. Chapter 3, "The user profile facility," on page 47 contains information about the PPRF command.

PMSG
Displays IMS messages that have been defined to exception analysis and have tripped.

Type:
Immediate

The PMSG command displays IMS messages that were defined with the MSGD command. The message must be defined with MSGD ADD and the OPTION=BUFSHOW or OPTION=BUFFER parameter.

Type the XIMS immediate command before the PMSG command to invoke the IMS message exception function and to keep current IMS messages displaying after the PMSG command.

PMSGnn
Displays IMS messages that have been defined to exception analysis and have tripped. If more than the specified number of priority messages exist, the command displays the most recent (nn) ones.

Type:
Immediate

The PMSGnn command displays IMS messages that were defined with the MSGD command. The message must be defined with MSGD ADD and the OPTION=BUFSHOW or OPTION=BUFFER parameter.

Type the XIMS immediate command before the PMSGnn command to invoke the IMS message exception function and to keep current IMS messages displaying after he PMSGnn command.

XACB
Lists and sets exception thresholds and attributes.

Type:
Immediate

Format:
XACB {ALL|GROUP=cc|LIST=cccc} {VERBOSE|TERSE} FORCE

(blank)
Lists all exceptions with their current settings in columnar display.

ALL
Lists all exceptions in invocation sequence. This is the default.

GROUP
Lists exceptions in group *cc*.

LIST
Lists exception *cccc*. To display 2 or more exceptions, type a blank between each *cccc* value.

VERBOSE
Use a multi-line display (the default) for each defined exception. Displays all exception parameters.

TERSE
Use a single-line display for each defined exception. Displays the following exception parameters:

Exception name
Threshold value
Display value
Exception state
Bell state

FORCE
This keyword causes the exception analysis routine to become active. If the exception trips based upon the current threshold settings, an exception message is displayed as if the exception analysis command is executing. To display a sample message text of a specific exception, the STATE=TEST Display parameter must be set for the exception.

XACB displays parameters in the following format:

```
 XACB LIST=cccc
: cccc
+  DISPLAY Parameters:      THRESHOLD Parameters:     XLF Parameters:
:    State=                  Threshold=                Auto=
:    Group=                  Display=                  Log=
:    Bell=                   Attribute=                Limit=nn (n)
+  BOX Parameters:          CYCLE Parameters:          Repeat=
:    Boxchar=' '             ExNcyc=n                  Persist=nn
:    Boxclr=                 Stop=n (m)                Sc=
:    Boxattr=                Cumulative=n
```

Figure 30. XACB parameters

To change an option value, type over the displayed value and press ENTER. The next time OMEGAMON for IMS invokes this exception, it uses these new characteristics.

The XACB multi-line display shown in Figure 30 on page 69 is comprised of the following parameters.

DISPLAY parameters

State
One of the following states:

NDSP
You can suppress the display of exceptions that you do not need to act on at this time. OMEGAMON for IMS treats the exception as ON, but the exception does not display. Instead, it can be logged to the XLFLOG, or can trigger an automatic screen space routine when it occurs.

ON
Invokes this exception during the current OMEGAMON for IMS session.

OFF
Does not invoke this exception during the current OMEGAMON for IMS session.

TEST
This parameter is used primarily for the purpose of training or demonstration. With the FORCE keyword, it causes a sample exception message to be displayed. When the TEST state forces a message to display, a *T* displays in column 2 of the message lines under XIMS.

Note: The zoom function is not available for exceptions in test mode.

Group
Specifies the two-character group identifier. IBM ships the product with groups predefined. Use the GDFN command to display existing groups or to define new groups. The settings for groups override the settings for individual exceptions.

Bell
Specifies whether the audible alarm on the terminal sounds when this exception occurs. The bell must be activated with the OPTN BELL=ON command.

THRESHOLD parameters

Threshold
Exception threshold. For exceptions that are just either ON or OFF and do not have a numeric threshold, this entry is blank.

Display
Sets the exception display color or intensity. Can be set to the variables Clr1 through Clr7. The variable values are defined with the .SCC command. They are associated with the following colors listed, and follow the same order.

Optionally, this value can be HI or LO on four- or non-color terminals and one of the following values on terminals that support the extended data stream.

RE
Sets the exception text red.

BL
Sets the exception text blue.

YE
Sets the exception text yellow.

PI
Sets the exception text pink.

GR
Sets the exception text green.

TU
Sets the exception text turquoise.

WH
Sets the exception text white.

NONE
Specifies the hardware default colors.

Note: The presentation of the intensity or color level on your terminal is determined by the type of terminal and the settings of the .SCC keywords.

Attribute
Sets an additional highlight attribute for the box.

BLINK
Turns on blinking for an exception.

RVRS
Displays message in reverse video.

UNDR
Underscores a message.

NONE
Uses the default extended highlight attributes.

These attributes take effect only in modes other than ISPF and cross memory/cross system on terminals that support the extended data stream.

XLF parameters
See Chapter 14, "OMEGAMON for IMS automation and logging features," on page 317 for an explanation of the exception logging facility (XLF) and automatic screen facility (ASF) parameters.

BOX parameters

Boxchar
Specifies box character, enclosed in single quotes. The default is a plus sign (+). Do not use a single quote as a box character, since it is the delimiter.

Enter NOBOX without quotes to turn off boxing for an exception. If Boxchar=NOBOX, then the Boxclr and Boxattr parameters have no effect.

Boxclr
Sets the color or intensity of the exception box.

The Boxclr= keyword settings follow the same format as the THRESHOLD parameter, Display=.

Boxattr
For seven-color terminals modes other than ISPF or cross-memory, sets an additional highlight attribute for the box.

BLINK
Turns on blinking for an exception.

RVRS
Displays message in reverse video.

UNDR
Underscores a message.

NONE
Specifies the hardware default attributes.

If you set Boxchar=NOBOX, then the Boxclr= and Boxattr= parameters have no effect. If you do not set color and highlighting attributes for the box, OMEGAMON for IMS uses those that you set for the exception.

CYCLE parameters

ExNcyc
Sets the frequency for checking the exception at every *n* OMEGAMON for IMS cycles. If this parameter is set to 0 or 1, it is tested every OMEGAMON for IMS cycle. If it is set to a higher number, it is tested only when that number of cycles elapses. The default setting for EXNCYC is 0.

This parameter is provided so that you can tailor high overhead exceptions for your own environment. You can avoid using CPU time to test them every cycle. For example, if you have many devices in the class that is examined by an exception (such as DASD or tape), you might want to set this parameter for corresponding exceptions in the hardware group.

When an exception that is not tested every cycle trips, the exception message displays on the screen as usual. In the following cycles during which it is not scheduled for testing, the exception message redisplays on the screen that follows the primary exception analysis display.

See also the `.NXE` immediate command that controls the display of frequency-limited exceptions.

Stop=n (m)
Sets a limit on the number of times an exception is allowed to trip. After the exception trips *n* times, the exception is not tested or does not display during the current OMEGAMON for IMS session, unless the user resets this parameter. The (*m*) value, which is informational only, indicates the number of times the exception already triggered since the user last reset the Stop parameter. The default value for Stop is 0, which means that there is no limit to how many times the exception can be tested and displayed.

Cumulative
Indicates how many times the exception triggered during the current OMEGAMON for IMS session. Users cannot alter this value.

The following figure shows a typical XACB display:

```
 XACB
: INAC
+      DISPLAY Parameters:   THRESHOLD Parameters:   XLF Parameters:
:        State=ON             Threshold=N/A           Auto=OFF
:        Group=IM             Display=CLR1            Log=OFF
:        Bell=OFF             Attribute=REVERSE       Limit=0
+      BOX Parameters:       CYCLE Parameters:        Repeat=NO
:        Boxchar='#'          ExNcyc=0                Persist=0
:        Boxclr=TURQUOISE     Stop=0 (0)              SS=
:        Boxattr=REVERSE      Cumulative=0
: DISP
+      DISPLAY Parameters:   THRESHOLD Parameters:   XLF Parameters:
:        State=ON             Threshold=N/A           Auto=OFF
:        Group=ST             Display=CLR1            Log=OFF
:        Bell=OFF             Attribute=REVERSE       Limit=0
+      BOX Parameters:       CYCLE Parameters:        Repeat=NO
:        Boxchar='#'          ExNcyc=0                Persist=0
:        Boxclr=TURQUOISE     Stop=5 (2)              SS=
:        Boxattr=REVERSE      Cumulative=2
```

Figure 31. XACB display

The following figure is a partial XACB terse mode display.

```
 XACB TERSE
: DNRS Threshold=N/A  Display=Red   State=ON     Bell=ON
: TNRS Threshold=N/A  Display=Blue  State=TEST   Bell=OFF
: WSHI Threshold=2500 Display=Pink  State=ON     Bell=OFF
: WSLO Threshold=300  Display=Blue  State=NDSP   Bell=OFF
```

Figure 32. XACB terse mode display

Controlling exceptions by groups

The XGSW command gives you control of exceptions by group.

XGSW

Sets exception group switch settings.

Use the group switch command to set the exception state for an entire exception group. This switch overrides the individual exception setting.

If you type in XGSW with no keywords, it displays all existing groups with their current settings. Type over the current setting for the STATE keyword to change the setting.

Type:

Immediate

Format:

XGSW GROUP=cc STATE=cccc

(blank)

Displays all existing groups with their current settings.

GROUP

Any two unique alphanumeric characters (*cc*) to specify the group. Use this keyword to display only the entries for a group.

STATE

Controls whether the exception is in any of these five states:

ON

Invokes the exception group during the current session.

OFF

Does NOT invoke the exception group during the current session.

TEST

Forces a sample warning message, even if the exception condition is not presently occurring, for purposes of training or demonstration. (When a message displays because of TEST mode, a **T** displays in column 2 of the message line.)

Note: The zoom function is not available for exceptions in test mode.

NDSP

Exceptions in the group are ON, but the exceptions are not displayed. Instead, they can be logged to the XLFLOG or can trigger automatic screen spaces.

NULL

By default, specifies that the individual exception, rather than the group switch, maintains control.

Resource contention exception commands

Use the resource contention immediate commands to display the exceptions that are caused by resource conflicts.

CONF

Displays IRLM or PI lock conflicts.

Type:

Immediate

Figure 33 on page 73 displays the CONF command output.

```
 CONF  Subsys   Workunit PSBname  TxRgID   Lterm ID Status DB/AREA   Token  DCB
+      IMSA     MPP00121 ACCNT010 UPDACCT  R105A10  UP/OWN ACCNTDBA 0000940C  1
+      IMSA     MPP00131 UPDCUST1 UPDCUST  L050C09  UP/WAT ACCNTDBA 0000940C  1
+      IMSA     MPP00132 UPDCUST1 UPDCUST  L050C15  UP/WAT ACCNTDBA 0000940C  1
```

Figure 33. CONF command output

The following list describes the output fields for CONF.

Subsys
Subsystem name that holds the lock.

Workunit
Name of job that holds the lock.

PSBname
Program specification block (PSB) associated with the lock.

Trxname
Transaction name that is associated with the lock.

Lterm ID
Logical terminal name that is associated with the lock.

An entry of NONE in this field indicates that there is no logical terminal that is associated with this lock.

Status
Intent and status of the workunit (job) holding the lock or waiting for the resource. Valid intents are as follows:

UP
Update intent.

RD
Read-only intent.

EX
Exclusive intent.

SH
Share intent.

ER
Erase intent.

Valid statuses are as follows:

OWN
Workunit owns this resource.

WAT
Workunit is waiting for this resource.

DB/AREA
Database name, DEDB area name, or partition name.

RBA/RBN
Relative byte address/relative block number. (Displays only if PI is used.)

DCB
Data control block (DCB) number within the named DMB.

DBName
Database name or DEDB area name.

PartName
Partition name (if partitioned).

Part.ID
: Partition ID (if partitioned).

n/a displays if data is not available because a lock is freed before analysis is complete.

Note: It is possible to see that a job is waiting for a resource when no owner for that resource is displayed, because the resource is locked by an IRLM running on another system.

XFPQ

Displays active fast path DEDB resource (control interval) request conflicts for all DEDB database areas.

Type:
: Immediate

Figure 34 on page 74 shows the results of the XFPQ command.

```
 XFPQ  Jobname  PSBName  Trxname  DBname   AREAname   R.B.A.   Status    P.I.
+      FPMSG1   DBFSAMP3 FPSAMP1  DBFSAMD3 CUSDB    0000C000  EX/OWNER    NO
```

Figure 34. XFPQ command

The following are the XFPQ fields and their meanings:

Jobname
: Name of region that is holding or waiting for resource. If name is OTHR, an output thread is holding the resource (control interval).

PSBName
: Name of the program specification block.

Trxname
: Name of the transaction.

DBname
: Name of the fast path DEDB.

AREAname
: Name of this partition of the DEDB.

R.B.A.
: Relative byte address - the address of the resource (control interval).

Status
: Status of the resource request. Options are as follows:

  EX/OWNER
  : exclusive/owner

  EX/WAITING
  : exclusive/waiting - highlighted

  NE/OWNER
  : non-exclusive/owner

  NE/WAITING
  : non-exclusive/waiting - highlighted

  P.I.
  : Does the IMS program isolation feature know the resource control request? YES or NO.

XLTQ

Scans the IMS latch tables and searches for conflicts.

Type:
: Immediate

When one dependent region holds an IMS latch that another region wants, the second region waits until the latch is available. A sample lockout of this type displays in Figure 35 on page 75:

```
 XLTQ  Jobname  PSBname  Trxname  Lterm ID   Status   Latch Type
+      BMP01    PY4BUP00 BPYUPD   --none--   Owner    LOGL  Logical Logger
+      MESSAGE  PY4MUP00 PAYRUP   L561       Waiting  LOGL  Logical Logger
+      ------------------------------------------------------------------------
+      J3       DBFSAMP3 FPSAMP1  L694       Owner    OPEN  FP DEDB Open
+      J2       DBFSAMP3 FPSAMP1  L652       Waiting  OPEN  FP DEDB Open
+      ------------------------------------------------------------------------
+      J2       DBFSAMP3 FPSAMP1  L652       Owner    DMAC  FP DEDB CUSDB
+      J1       DBFSAMP3 FPSAMP1  L698       Waiting  DMAC  FP DEDB CUSDB
```

Figure 35. Sample lockout

In this example, the region BMP01 holds the logical logger latch (LOGL) and the region MESSAGE wants it; MESSAGE waits until BMP01 frees the latch.

Chapter 5. IMS system information

You can use OMEGAMON for IMS commands to display IMS system information.

Real memory information

Use the CSAR command to display real memory information.

CSAR

Displays the users of CSA real memory by their storage protect keys.

Type:

Immediate

IMS CSA is in storage key 7.

If you run more than one IMS system at a time, the real memory for key 7 reflects the CSA usage of all your IMS systems and not just the one thatOMEGAMON for IMS is monitoring.

Common storage area virtual memory analysis

You can use the CSA, ECSA, ESQA, and SQA commands to analyze common storage area virtual memory.

CSA

Analyzes the common storage area by subpool number and protection key.

Type:

Immediate

ECSA

Analyzes the extended CSA by subpool number and protection key.

Type:

Immediate

ESQA

Analyzes the extended system queue area.

Type:
Immediate

The ESQA command displays the amount of virtual memory that is allocated and used for each subpool. The Ext field displays the number of extents that are assigned to subpools. (A GETMAIN request that z/OS cannot satisfy with the storage currently allocated to a subpool causes z/OS to assign another extent to the subpool: these extents are always multiples of 4K long.) The allocated blocks field displays the amount of virtual storage that is assigned to a subpool. The storage used field shows the amount of space within the storage that is assigned to a subpool that is in use: that is, turned over to the user in response to a GETMAIN request. The largest free block is the size of the largest chunk within a subpool that is not currently in use, except that the total line shows the largest block that is not allocated to a subpool.

The storage values display in megabytes + kilobytes + bytes. A megabyte (M) is 1,048,576 bytes, while a kilobyte (K) is 1024 bytes.

OMEGAMON for IMS cannot break down the usage of ESQA by address space because, in general, the operating system does not save this information. The following figure shows a typical SQA and ESQA utilization display.

```
SQA   Subpl-Key  Ext     Alloc. Blks        Storage Used     Largest Free Blk
 +       226-0      4             56K            52K+128              2K
 +       239-0      3             88K            80K+816              3K+1,016
 +       245-0      6            368K           185K+48             168K+72
 +                ---   -------------  -------------------
 +      Total:     13            512K           317K+992
 +      Percent of SQA:        100.0%          62.1%
 +      SQA Size:                512K
  ESQA  Subpl-Key  Ext     Alloc. Blks        Storage Used     Largest Free Blk
 +       239-0      3             52K            48K+184              3K+736
 +       245-0      4         8M+204K           479K+208          7M+736K+592
 +                ---   -------------  -------------------  -------------------
 +      Total:      7         8M+256K           527K+392
 +      Percent of ESQA:       100.0%           6.2%
 +      ESQA size:            8M+256K
```

Figure 36. SQA and ESQA utilization display

SQA

Analyzes the system queue area.

Type:
Immediate

The SQA immediate command displays virtual memory usage by subpool and protect key.

The SQA command displays the amount of virtual memory that is allocated and used for each subpool. Ext displays the number of extents that are assigned to subpools. (A GETMAIN request that z/OS cannot satisfy with the storage currently allocated to a subpool causes z/OS to assign another extent to the subpool: these extents are always multiples of 4K long.) The allocated blocks field displays the amount of virtual storage that is assigned to a subpool. The storage that is used is the amount of space within the storage that is assigned to a subpool that is in use: that is, turned over to the user in response to a GETMAIN request. The largest free block is the size of the largest chunk within a subpool that is not currently in use, except that the total line shows the largest block that is not allocated to a subpool.

The storage values display in megabytes + kilobytes + bytes. A megabyte (M) is 1,048,576 bytes, while a kilobyte (K) is 1024 bytes.

OMEGAMON for IMS cannot break down the usage of SQA by address space because, in general, the operating system does not save this information.

CSA storage isolation information

You can use the CSTI command to display information about CSA storage isolation.

CSTI
Displays the parameters and data relevant to CSA storage isolation.

Type:
Immediate

For CSA storage isolation, paging is fenced in units of page-ins/second (rather than page-ins/CPU-second for address space fencing). In Figure 37 on page 79, the fence limits are set at two and five page-ins per second: the current page-ins rate is one CSA page/second.

```
 CSTI  Common:      Working         Page-ins
+                   Set Size        /second
+      Maximum       1520K               5
+      Target        1500K          ------
+      Actual        1500K               1
+      Minimum        200K               2
```

Figure 37. CSA storage page-ins

IPS parameters are specified in 4K blocks, rather than in units of 4K, as the previous example shows. The common area consists of CSA + PLPA.

When you specify storage isolation for IMS, bear in mind that all cross-memory address spaces use page-ins per second as opposed to page-ins per execution second.

If you use the DLISAS address space, the IMS control region is not a cross-memory address space. Use execution seconds when you calculate storage isolation for the control region.

Application Trace Facility (ATF) commands

You can access the ATF commands through the menu interface.

The ATF commands are as follows:

ATFL
Filter application trace displays.

ATFO
Set ATF trace default profile options.

ATMN
Manage application trace.

ATSP
Specify application trace.

ATVD/ATXD
View application trace details.

ATVG
View application trace summary by group.

ATVS/ATXS
View application trace summary.

ATVW/ATXW
View application trace overview.

ATVX/ATXX
View application trace DL/I details.

IMS control block display commands

You can use the control block display commands to dump various IMS control blocks and tables.

The control block display commands are as follows:

DUMP
Groups together a set of minor commands, which dump various IMS control blocks.

Type:
Major

By default, each minor produces a display in hexadecimal format, where each line is 16 hex bytes followed by its character equivalents.

Note: Each minor also accepts either *X* or *C* in the label field, if you want the display to be all hexadecimal or all character.

$DIR
Dumps the $$IMSDIR table.

Type:
Minor of DUMP

Format:
$DIRcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of an entry.

BALG
Dumps balancing group *nn*.

Type:
Minor of DUMP

Format:
BALGnn

nn
Specifies the control block to dump.

BCPT
Dumps the checkpoint ID table.

Type:
Minor of DUMP

Format:
BCPT

BFSP
Dumps the VSAM buffer pool prefix.

Type:
Minor of DUMP

Format:
BFSP

BFUS
Displays the VSAM buffer statistics block.

Type:
Minor of DUMP

Format:
BFUSnn

nn
Specifies the control block to display.

BHDR
Displays the main storage database header.

Type:
Minor of DUMP

Format:
BHDRcccccccc

cccccccc
Specifies a number, *n*. to dump the *n*th control block entry or the name of an MSDB.

BSPH
Displays the VSAM buffer subpool header.

Type:
Minor of DUMP

Format:
BSPHnn

nn
Specifies the control block to display. This number corresponds to the VSAM subpool number the DBVS immediate command displays.

CCB
Dumps the specified conversational control block.

Type:
Minor of DUMP

Format:
CCBnn

nn
Specifies the control block to display.

CLB
Dumps the communications line block.

Type:
Minor of DUMP

Format:
CLBcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a CLB.

CNT
Dumps the specified communications name table.

Type:
Minor of DUMP

Format:
CNTcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a CNT.

CTB
Dumps the communications terminal block.

Type:
Minor of DUMP

Format:
CTBcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a CTB.

DDIR
Dumps the DMB directory entry.

Type:
Minor of DUMP

Format:
DDIRcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a DMB.

DMAC
Dumps the DEDB area control.

Type:
Minor of DUMP

Format:
DMACcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a DEDB.

DMCB
Dumps the DEDB master control.

Type:
Minor of DUMP

Format:
DMCBcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a DEDB.

DMHR
Dumps Fast Path buffer headers.

Type:
Minor of DUMP

Format:
DMHRnn

nn
Specifies the control block to dump.

ECNT
Dumps extended CNT.

Type:
Minor of DUMP

Format:
ECNTcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of an LTERM.

EPST
Dumps extended PST.

Type:
Minor of DUMP

Format:
EPSTcccccccc

cccccccc
Specifies a number (*n*) to dump the *n*th control block entry or the job name of a region.

ESCD
Dumps extended SCD.

Type:
Minor of DUMP

FIDX
Dumps the MFS incore directory index.

Type:
Minor of DUMP

Format:
FIDXcccccccc

cccccccc
Specifies a number (*n*) to dump the *n*th control block entry or the name of an entry.

IBPL
Dumps the ISAM/OSAM buffer pool prefix.

Type:
Minor of DUMP

IBPR
Dumps the ISAM/OSAM buffer subpool buffer prefix.

Type:
Minor of DUMP

Format:
IBPRnn

nn
Specifies the subpool to dump.

IPB
Dumps the initialization parameter block.

Type:
Minor of DUMP

ISBP
Dumps the ISAM/OSAM buffer subpool header.

Type:
Minor of DUMP

Format:
ISBPnn

nn
Specifies the control block to dump. This number corresponds to the ISAM/OSAM subpool number the DBOS immediate command displays.

MFBP
Dumps the message format buffer pool header.

Type:
Minor of DUMP

MLCD
Dumps the DC monitor log control directory.

Type:
Minor of DUMP

PAB

Dumps the parameter anchor block.

Type:
Minor of DUMP

PDIR

Dumps the PSB directory entry.

Type:
Minor of DUMP

Format:
PDIRcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a PSB.

PST

Dumps the partition specification table.

Type:
Minor of DUMP

Format:
PSTcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the job name of the region.

QBUF

Dumps the message queue buffer pool header.

Type:
Minor of DUMP

RCTE

Dumps route codes entry.

Type:
Minor of DUMP

Format:
RCTEcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of a route code entry.

RSIN

Dumps the checkpoint/restart log record buffer.

Type:
Minor of DUMP

Format:
RSINnn

nn
Specifies the control block to dump.

SAP

Displays the save area set prefix.

Type:
Minor of DUMP

Format:
SAPnnn

nnn
Specifies the control block to dump.

SCD
Dumps the system contents directory.

Type:
Minor of DUMP

SLCD
Dumps the system log control directory.

Type:
Minor of DUMP

SMB
Dumps the scheduler management block.

Type:
Minor of DUMP

Format:
SMBcccccccc

cccccccc
Specifies a number, *n*, to dump the *n*th control block entry or the name of an SMB.

SPID
Displays the subpool ID table.

Type:
Minor of DUMP

Format:
SPIDnn

nn
Specifies the first subpool ID control block number. You must enter this number, as there might be multiple tables chained together.

SPQB
Dumps the subpool queue control block.

Type:
Minor of DUMP

Format:
SPQBcccccccc

cccccccc
Specifies a number *(n)* to dump the *n*th control block entry or the name of an SPQB.

The following figure shows an example of the DUMP major command and some of its minors.

```
DUMP  Display IMS Control Blocks
 ddir  DI21PART
  ADDR=B4DE20
+ 000  00B4DDAC 8A000000 C4C9F2F1 D7C1D9E3  *........DI21PART*
+ 010  009207A0 00030C00 00640064 FFFF0006  *.k..............*
+ 020  40200000 00000000                    * .......        *
================================================================================
Xddir  DI21PART
  ADDR=B4DE20
+ 000  00B4DDAC 8A000000 C4C9F2F1 D7C1D9E3 009207A0 00030C00 00640064
FFFF0006
+ 020  40200000 00000000
================================================================================
Cddir  DI21PART
  ADDR=B4DE20
+ 000  *........DI21PART.k.............. .......
      *
================================================================================
 sap  1
  ADDR=B50900
+ 000  D2880000 00000000 009721EC 00972234  *Kh.......p...p..*
+ 010  009721C0 00B4C588 00000000 00000000  *.p.{..Eh........*
+ 020  00000007 00000010 00000010 009042E0  *...............\*
+ 030  00000000 00000000 00000000 00000000  *................*
+ 040  00000000 00000000 00000000 00000000  *................*
+ 050  00000000 00000000 00000000 00000000  *................*
+ 060  00000000 00000000 00000000 00000000  *................*
+ 070  00000000 00000000 00000000 00000000  *................*
```

Figure 38. DUMP major command

Dynamic control block table analysis

You can use the CBTA, CBTL, and CBTP commands to display data about dynamic storage area control block tables (IPAGES).

For a list of the current dynamic storage area control block tables, see the IMS.GENLIBB member DFSCBTS.

CBTA
: Selects all dynamic storage area control block tables.

  Type:
  : Major

  IPGA is an alias for the CBTA command.

CBTL
: Selects listed dynamic storage area control block tables.

  Type:
  : Major

  IPGL is an alias for the CBTL command.

CBTP/n
: Selects all dynamic storage area control block tables that match a pattern.

  Type:
  : Major

  Format:
  : **CBTP/n**

    /n
    : Specifies the pattern

  The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If CBTP does not find a /n value, it uses the first pattern that is supplied with the last **.SPT** command.

  IPGP is an alias for the CPTP command.

Dynamic control block table minor commands

For more information about dynamic storage area control block tables (IPAGES), use the minor commands of CBTA, CBTL, CBTP.

The following example shows CBTA, CBTL, and CBTP minor commands.

```
CBTA      IOSB     GIOB     OSWA     GOWA      PST     DPST      SAP     GQMW
blkl       576        1     1024     1024     2312     2312      128      256
blkn        21                 7               12        2       94       30
blku         4                                 11        1       76       16
curr     12288              8192            49152     8192    12288     8192
fres
gets         1                 1               12        2        2        2
ipfr                           1                         1
ipln     12288              8192             4096     4096     8192     4096
locp   CSA/CTL  CSA/CTL  CSA/CTL  CSA/CTL  CSA/CTL  CSA/CTL  CSA/CTL  CSA/CTL
maxs     12288              8192            49152     8192    12288     8192
npgs         1                 1               12        2        2        2
sbpl       228      228      228      228      231      231      231      231
```

Figure 39. CBTA, CBTL, and CBTP minor commands

BLKL
: Displays length of the control block for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

BLKN
: Displays total number of control blocks for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

BLKU
: Displays total number of control blocks that are used for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

CURR
: Displays CURRENT STORAGE usage for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

FRES
: Displays total number of FREEMAINs for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

GETS
: Displays total number of GETMAINs for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

IPFR
: Displays total number of free IPAGES.

  Type:
  : Minor of dynamic control block table majors

IPLN
: Displays the IPAGE length for this IPAGE type.

  Type:
  : Minor of dynamic control block table majors

LOCP
Displays location of the control block.

Type:
Minor of dynamic control block table majors

The location is CSA or PVT (private) and CTL (control region) or DLS (DLISAS region).

MAXS
Displays MAXIMUM STORAGE usage for this IPAGE type.

Type:
Minor of dynamic control block table majors

NPGS
Displays TOTAL NUMBER of IPAGEs for this IPAGE type.

Type:
Minor of dynamic control block table majors

SBPL
Displays z/OS SUBPOOL for this IPAGE type.

Type:
Minor of dynamic control block table majors

Display checkpoint ID information

Use the CKPT command to display the current checkpoint ID and the IMS log volume serial numbers.

CKPT
Displays general IMS checkpoint information.

The following figure shows an example of CKPT output.

```
CKPT  Current Checkpoint id = 2011067185504577
+      Block Number = 5
+      -----------------------------------------------------------------------
+      Latest DUMPQ/SNAPQ Checkpoint id = 2011024210057483
+      Block Number = 2
```

Figure 40. CKPT output

MFS format blocks in memory

Use the **MFS** major commands to display information about MFS format blocks that are in memory.

FMTI
Selects all in-memory FREs.

Type:
Major

FMTL
Selects in-memory FREs listed.

Type:
Major

FMTP/n
Selects all in-memory FREs by pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If FMTP does not find a /n value, it uses the first pattern that is supplied with the last **.SPT** command.

The following minor commands display the type, size, address, and status of MFS format blocks that are in memory.

FADR
Displays address of the FRE.

Type:
Minor of MFS format block majors

MADR
Displays address of associated MFS format blocks.

Type:
Minor of MFS format block majors

NUSR
Displays number of immediate fetch and pre-fetch users for the displayed format.

Type:
Minor of MFS format block majors

SIZE
Displays size of associated MFS format blocks.

Type:
Minor of MFS format block majors

STAT
Displays status of the associated MFS format blocks.

Type:
Minor of MFS format block majors

The status of the format block is as follows:

Free Blk
Block is free for reuse.

ImmFetch
Block is on the immediate-fetch chain.

-in mem-
The MFS format is currently in memory.

-in use-
Block is currently in use.

-not fnd-
Block not found in memory.

PreFetch
Block is on the pre-fetch chain.

TYPE
Displays type of associated MFS format blocks.

Type:
Minor of MFS format block majors

Type is DIF/DOF or MID/MOD.

The example that follows shows the results of the FMTL command with the SIZE, TYPE, and STAT minor commands.

```
FMTL    DFSMO1    DFSMO2    DFSMI2    DFSMI1
 size        60       186       122        80
 type   MID/MOD   MID/MOD   MID/MOD   MID/MOD
 stat  Free Blk -not fnd-  Free Blk  -in use-
```

Figure 41. FMTL command display

Display information from $$IMSDIR table

The $$IMSDIR table is a special index for the MFS data set. When IMS uses the $$IMSDIR table, it reduces format block I/O. The pool manager can do a direct read from the MFS data set instead of scanning the IMS.FORMAT PDS directory.

IMS stores the $$IMSDIR table in the MFS buffer pool. The $$IMSDIR table entries initialize the MFS dynamic directory, which is in extended private storage. After initialization, the IMS control region automatically adds entries to the MFS dynamic directory as format blocks are fetched from the format libraries.

The output of the following commands displays information from the MFS dynamic directory, rather than the original $$IMSDIR table entries.

DIRA
Selects all entries in $$IMSDIR.

Type:
Major

DIRL
Selects $$IMSDIR entries listed.

Type:
Major

DIRP/n
Selects all $$IMSDIR entries that match a pattern.

Type:
Major

The `.SPT` command sets the pattern. See the `.SPT` command for information about setting patterns. If DIRP does not find a /n value, it uses the first pattern that is supplied with the last `.SPT` command.

DIRU
Selects $$IMSDIR entries, which are unusable.

Type:
Major

These entries are entries that IMS does not find in the IMS.FORMAT data set.

The following minor commands display the type and size of the MFS format blocks.

SIZE
Displays size of associated MFS format blocks.

Type:
Minor of $$IMSDIR table majors

TYPE
Displays type of associated MFS format blocks.

Type:
Minor of $$IMSDIR table majors

The format type is DIF/DOF or MID/MOD.

The following figure shows the results of the DIRL command with the SIZE and TYPE minor commands.

```
DIRL     DFSMO1   DFSMO2   DFSMI2   DFSMI1
size         60      186      122       80
type    MID/MOD  MID/MOD  MID/MOD  MID/MOD
```

Locate IMS modules in virtual memory

Use the FCDEcommand to locate IMS modules in virtual memory.

FCDE

Locates IMS modules in virtual memory.

Type:

Immediate

The FCDE immediate command accepts an eight-character IMS module name as input. It then searches the IMS jobpack queue and the PLPA directory to locate the contents directory entry (CDE) for the module.

OMEGAMON for IMS also searches the jobpack queue for all IMS system control address spaces (including DLISAS, DBRC, and IRLM).

If OMEGAMON for IMS finds the module, it displays various information about it, as the following figure shows. You can locate various IMS control blocks, since IMS artificially creates CDEs for many of its important work areas. The following figure shows an FCDE command display.

```
FCDE  DFSMVRC0
 +      Module located in the CTL Region Jobpack Queue
 +      Entry Point = 00115E78  Length = 000188  Use Count = 1
 +      ------------------------------------------------------------------
 +      Module located in the DLS Region Jobpack Queue
 +      Entry Point = 00115E78  Length = 000188  Use Count = 1
 +      ------------------------------------------------------------------
 +      Module located in the DBRC Region Jobpack Queue
 +      Entry Point = 00115E78  Length = 000188  Use Count = 1
```

Figure 42. FCDE command display

IMS startup parameters and overrides

Use the FCPB command to display the startup parameters for an IMS control region.

FCPB

Displays IMS control region startup parameters and indicates whether the displayed parameter value is user-initiated or is overridden by IMS initialization processing.

Type:

Immediate

You can override IMS control region options with the PARM= keyword of the EXEC card and with the DFSPRRGx member, found by using a normal STEPLIB search. The specified options merge according to rules in the *IMS/VS System Programmer's Reference Manual*.

Various symbols might display next to the displayed value. Their meanings are as follows.

(blank)

IMS uses the value that is specified in the IMS SYSGEN. Unspecified sysgen values might be defaulted by IMS. These parameters have a blank after them.

*****

IMS uses the value that is specified in the startup JCL or the DFSPRRGx parm member.

******

IMS determines that the value that is specified in the JCL, in DFSPRRGx, or in the SYSGEN is inadequate and therefore provides a new value.

||

The specified parameter is obsolete.

When no characters are displayed next to the value, the value is specified in the IMS SYSGEN.

If a DFSDRFnn member in PROCLIB, with a BLOCKS=VTCB statement, is selected at IMS initialization, the FCPB display indicates that the VTCB DREF option is specified.

The following figure is a sample FCPB display.

```
 FCPB  ------ Control Region Specifications ------------------------------------
+        RES=    Y       FRE=   30*     QBUF=    5*      PST=    1*
+        SAV= 19,500    EXVR=    1*      PRF=     ||    SRCH=    0*
+        SOD=    0       IOB=     ||    VAUT=    1*     FMTO=    D*
+       AUTO=    N*      TRN=    N*      SGN=    N*      RCF=    N*
+      IMSID= I41A      ISIS=    0*     NLXB=    0       LSO=    S
+       DBRC=    Y||    IRLM=    N    IRLMNM=            SSM= SSM *
+       WADS=    D*      ARC=    1      FBPR=     ||   UHASH= DBFLHSH0
+        QTU=   75*      QTL=   50*   DBRCNM=IMS500AB* DLINM= IMS500AD*
+      HSBID=           RECA=    5*     RECH=     ||  FESTIM=    0*
+        CRC=  ‘ ‘      APPC=    N*      ETO=    Y*     DLQT=   60*
+       ASOT=   60*     ALOT=   60*
+      ------ Storage Pool Sizes in 1K blocks ----------------------------------
+        FBP=   48**     PSB=   12**     DMB=   48**  DLIPSB=   40*
+       TPDP=     ||    WKAP=   48**    PSBW=   24**    SPAP=     ||
+       DBWP=   24**     MFS=     ||  CSAPSB=   12*
+      ------ Expandable Storage Pool Upper Limits in bytes -------------------
+       CIOP=   2G*     SPAP=   2G*     LUMC=   2G*     LUMP=   2G*
+       HIOP=   2G*     CESS=   2G*     EMHB=   2G*     FPWP=   2G*
+      ------ Member Suffixes -----------------------------------------------
+        SUF=  ‘0’       FIX= ‘DC’*     PRLD= ‘DC’*    VSPEC= ‘DC’*
+     HSBMBR= ‘00’*      SPM= ‘  ‘
+      ------ Fast Path Specifications ----------------------------------------
+       BSIZ= 2048*     OTHR=    5*     DBFX=   10*     DBBF=   50*
+       MSDB=  ‘C’*     LGNR=   10*     EPCB=    8**
+
+      blank = value used in IMS gen    *  = value used in startup parms
+      **    = IMS overrode all values  || = parameter is obsolete
```

Figure 43. FCPB sample display

The following figure shows a sample FCPB display when you monitor IMS Version 11, or higher versions.

```
 FCPB  ------ Control Region Specifications ------------------------------------
+        RES=    Y       FRE=   30*     QBUF=    5*      PST=    5*
+        SAV= 18**,500  EXVR=    1*      PRF=     ||    SRCH=    0*
+        SOD=    0       IOB=     ||    VAUT=    1*     FMTO=    D*
+       AUTO=    N*      TRN=    N*      SGN=    N*      RCF=    N*
+      IMSID= IB1G*     ISIS=    0*     NLXB=    0       LSO=    S
+       DBRC=    Y||    IRLM=    Y*   IRLMNM= IRBG*      SSM= SSM *
+       WADS=    D*      ARC=    1      FBPR=     ||   UHASH= DBFLHSH0
+        QTU=   75*      QTL=   50*   DBRCNM=IMSB10GB* DLINM= IMSB10GD*
+      HSBID=           RECA=    5*     RECH=     ||  FESTIM=    0*
+        CRC=  '/'*     APPC=    Y*      ETO=    Y*     DLQT=   60*
+       ASOT=   60*     ALOT=   60*    PTYPE=CACHE64|$ PNAME=ACBIN64|$
+      ------ Storage Pool Sizes in 1K blocks ----------------------------------
+        FBP=   48*      PSB=   12**     DMB=   48*   DLIPSB=   40*
+       TPDP=     ||    WKAP=   48*     PSBW=   48*     CWAP=     ||
+       DBWP=   24*      MFS=     ||  CSAPSB=   12*
+      ------ Expandable Storage Pool Upper Limits in bytes -------------------
+       CIOP=   2G*     SPAP=  n/a      LUMC=   2G      LUMP=   2G
+       HIOP=   2G*     CESS=   2G      EMHB=   2G*     FPWP=   2G*
+      ------ Member Suffixes -----------------------------------------------
+        SUF=  'G'*      FIX= 'DC'*     PRLD= ' C'*    VSPEC= 'BG'*
+     HSBMBR= 'DC'*      SPM= '  '      CSLG= 'ECE'*   DFSDF= '001'*
+      ------ Fast Path Specifications ----------------------------------------
+         FP=    Y*    FPOPN=    N*    FPRLM=    N*  POOLSIZ=  25|$
+       EMHL=  256*   SVSODR= NONE* MXOVFLSZ=    6|$
+
+      blank = value used in IMS gen    *  = value used in startup parms
+      **    = IMS overrode all values  || = parameter is obsolete
+      |$    = v11 64bit include dynamic acblib alloc
 ===============================================================================
```

Figure 44. FCPB sample display for monitoring IMS Version 11, or later

IRLM startup parameters

Use the RLMO and RLMX commands to display the IRLM information.

RLMO

Displays IRLM startup options.

Type:

Immediate

```
 RLMO  ----  IRLM options  -----------------------------------------------------
+       IRLMNM=     KRLM    IRLMID=      1  COMCYCL=      20    MAXCSA= 1(  1M)
+      DEADLOK='15,4'        SCOPE=  LOCAL    RULES=  COMPAT        PC=    YES
+     -----  Appl options  -----------------------------------------------------
+       APPLS= KRLM1,PRLM1                    APPL2= KRLM1,TRLM1
+       APPL3= KRLM1,BRLM1
+     -----  IRLM Trace status  ------------------------------------------------
+          RH=     ON        PTB=    OFF  INTERNAL=     OFF
```

Figure 45. RLMO sample display

A description of the fields in the RLMO display follows. Unless indicated, these fields represent the parameter values specified in the IRLM startup procedure.

Note: The parameters that display on your panel might be different from this example, depending on which version of IRLM your site is using.

IRLMNM=
Specifies the IRLM subsystem name.

IRLMID=
Displays the ID= parameter.

COMCYCL=
Displays the COMCYCL= parameter.

MAXCSA=
Displays the MAXCSA= parameter and the actual storage size.

CURRENT=
Displays the current CSA usage.

DEADLOK=
Displays the DEADLOK= parameter.

SCOPE=
Displays the SCOPE= parameter (GLOBAL or LOCAL).

RULES=
Displays the RULES= parameter (COMPAT or AVAIL).

PC=
Displays the PC= parameter (YES or NO).

APPLS=
Displays the APPLS= parameter

APPL2=
Displays the APPL2= parameter.

APPL3=
Displays the APPL3= parameter.

RH=
Displays if request handler trace is active or inactive (ON or OFF).

PTB=
Displays if pass-the-buck trace is active or inactive (ON or OFF).

INTERNAL=
Displays if the internal trace is active or inactive (ON or OFF).

RLMX

The RLMX command displays IRLM contention and coupling facility information as shown in the following figure.

Type:
Immediate

```
RLMX
  Lock Structure
           Name . . . . . . . . . . . : IRLMLT1
           Size . . . . . . . . . . . :  8388608
        Record List Entries  (RLEs)
           Used . . . . . . . . . . . :      871
           In Lock Structure  . . . :    44719
           Percentage Used  . . . . :     1.95%
       Totals:
          Real Contention Granted . :     2118
          Real Contention Rate  . . :      5.6/s
          Global False Contention . :      721
          False Contention Rate . . :      .67/s

      N-way Datasharing Subsystems for Group IRLMD1

      CF Name    Jobname    System ID    Status
      --------   --------   ---------    --------
      IRLMD1$$   IMS510AI   SP11         Active
      IRLMD1$$   IMS510NI   SP13         Active
```

Figure 46. RLMX command display

Note: This command applies to IRLM 2.1 and higher versions and to z/OS 5.1, 5.2, and OS/390®.

This command displays statistical information about IRLM real and false contention.

Real contention occurs when two PSBs attempt to access the same database block at the same time. False contention occurs when there is a lock synonym, but no real contention.

The Real Contention Rate is the actual lock rate, per second, during this interval. False Contention Rate is the lock synonym rate, per second, during this interval.

In a z/OS 5.1, 5.2, or OS/390 environment, you can define your IRLM to participate in a Data Sharing group. The group name you defined is also displayed plus all of the participants in the Data Sharing group.

CF Name
Coupling Facility name.

Job Name
The z/OS started task or job name of the participating IRLM subsystem.

System ID
The z/OS System ID on which the IRLM subsystem is running.

Status
The current Coupling Facility status.

ACTIVE
The IRLM subsystem is actively connected to the Coupling Facility and is eligible for N-Way Data Sharing.

CREATED
The IRLM subsystem is defined to the Coupling Facility but is not yet active. This state is a transient state.

FAILED
The IRLM subsystem failed to connect to the Coupling Facility. See your z/OS system console for error messages.

QUIESCED
The IRLM subsystem is removing itself from the Coupling Facility. This state is a transient state.

IMS control region virtual storage analysis

You can use the VMEM command to track virtual storage usage.

VMEM
Tracks virtual storage use and warns when shortages develop within the IMS control region.

Type:
Immediate

Exception analysis monitors various virtual storage areas for exceptional conditions, which are based on thresholds you set. With the VMEM command you can identify storage shortages, as well as excesses. To trim these excesses back, you can readjust the position of IEALIMIT within the control region. The following figure is an example of a typical VMEM command display.

```
 VMEM  IMS Virtual Storage Constraint Analysis
+
+                           Lsqa    Lsqa     Lsqa     Private Private  Private
+                           Free    Largest  Assured  Free    Largest  Top block
+      Control Region :     5836k    5796k    3700k    2288k    2136k    2136k
+      DBRC    Region :     7224k    7220k    4892k    2356k    2332k    2332k
+      DLS     Region :     7428k    7424k    3868k    3592k    3560k    3560k
+      IRLM    Region :     7896k    7884k    5924k    1972k    1972k    1972k
```

Figure 47. VMEM command display

A diagram of IMSCTL region virtual storage areas follows.

```
      +-------------------+
      |                   |
      |      LSQA         |
      |                   |      Current
      |       ------------|<---  LSQA      <----+
      |      /  Assured   |      Position       |
      |     /   Free      |                     |  Block free
      |    /    LSQA      |                     |  at top of
      |-_-/_-_-_-_-_-_-_-_| =IEALIMIT              private area
      |  / Top block      |                     |  can span
      | /  free within    |                     |  IEALIMIT and
      |/   IMSCTL private |      Current        |  is also
      |-------------------|<---  top of   <-----+  available
      |     IMS           |      IMSCTL            for LSQA
      |     Control       |      region
      |     Region        |      private
      |     private       |      area
      |     area          |
      +-------------------+

 1.LSQA can acquire free space below the IEALIMIT line.
 2.The control region private area can acquire space up to the IEALIMIT line.
 3.If LSQA has extended BELOW the IEALIMIT line, then the control region cannot extend above
 LSQA's
 present position, otherwise an S40D abend, or similar shortage abend, might occur.
```


```
      +-------------------+
      |                   |
      |      LSQA         |
      |                   |      Current
      |       ------------|<---  LSQA      <----+
      |      /  Assured   |      Position       |
      |     /   Free      |                     |  Block free
      |    /    LSQA      |                     |  at top of
      |-_-/_-_-_-_-_-_-_-_| =IEALIMIT              private area
      |  / Top block      |                     |  can span
      | /  free within    |                     |  IEALIMIT and
      |/   IMSCTL private |      Current        |  is also
      |-------------------|<---  top of   <-----+  available
      |     IMS           |      IMSCTL            for LSQA
      |     Control       |      region
      |     Region        |      private
      |     private       |      area
      |     area          |
      +-------------------+

 1.LSQA can acquire free space below the IEALIMIT line.
 2.The control region private area can acquire space up to the IEALIMIT line.
 3.If LSQA has extended BELOW the IEALIMIT line, then the control region cannot extend above
 LSQA's
 present position, otherwise an S40D abend, or similar shortage abend, might occur.
```


Figure 48. IMSCTL region virtual storage areas

OMEGAMON for IMS monitors the following virtual storage quantities by request or dynamically through exception analysis.

- Total free storage available for LSQA
- Largest free block available for LSQA
- Amount of free storage above the IEALIMIT for LSQA: (assured free LSQA)
- Total free storage available for IMSCTL
- Largest free block available for IMSCTL
- Size of free block at top of IMSCTL region

OMEGAMON for IMS analyzes deallocates only free areas, and not free areas within allocated storage. It excludes these small areas, which result from fragmentation.

IMS status information

Use the **ASYS** command to display information about the status of the APPC/IMS system. Use **ISYS** to display general information about the status of the IMS system and use **MSYS** to display general information about the status of the z/OS system

ASYS
Displays information about the state of the APPC/IMS system.

Type:
Immediate

Format:
ASYS

ASYS displays the following information.

- APPC/IMS status summary
- APPC/IMS LU6.2 descriptors

If you have IMS without Remote Site Recovery installed, the system displays ASYS information as shown in the following example:

```
ASYS    IMS Base Luname = IMSLU91C      IMS Base Network Id = USCAC001
 +       APPC Status = ENABLED          APPC Desired Status = ENABLED
 +       RACF Option = NONE             Generic Resources LU = --N/A--
 +       LU62 Active Sync Convs = 0     LU62 Active Async Convs = 0
 +
 +       DESCRIPTOR SIDE  NETID   LUNAME    MODE      TYPE   SYNCLVL
 +       >>> LU62  Descriptors not defined
```

Figure 49. ASYS information for IMS without remote site recovery

The fields in the ASYS display are explained in the following list.

IMSLU
This identifies the base IMS LU name. The LU name displays N/A if IMS is not connected to APPC/z/OS. N/A applies for DISABLED, STARTING, or FAILED status.

IMSLU Netid
The network ID for the IMSLU. NETID displays N/A if IMS is not connected to APPC/z/OS. N/A applies for DISABLED, STARTING, or FAILED status.

APPC Status
The current IMS status of the APPC connection.

APPC Desired Status
The required IMS status of the APPC connection.

RACF® Option
The RACF security level. FULL (default) is the RACF check a clone ACEE to the dependent region. CHECK completes an RACF check, but not the cloning of ACEE. NONE indicates that no RACF calls are completed in IMS. PROFILE is the security option in the TP Profile.

Generic Resources Name
APPC generic LU name. N/A displays if VTAM Generic Resources are not active.

Active Sync Convs
Number of active LU 6.2 synchronous conversations.

Active Async Convs
Number of active LU 6.2 asynchronous conversations. This display shows an LU 6.2 descriptor entry values for LU 6.2 descriptors if they are defined in IMS.

DESC
Descriptor name.

LUNAME
LUNAME for the descriptor. Not valid when SIDE is specified.

SIDE
APPC/z/OS side table for the descriptor. SIDE value overrides LUNAME and MODE entries.

SYNCLVL
Synchronous level (either CONFIRM or NONE).

TYPE
Conversation type (either BASIC or MAPPED).

MODE
VTAM mode table entry for descriptor, which is not valid when SIDE is specified.

TPNAME
TP name for descriptor.

ISYS
Displays general information about the state of the IMS system.

Type:
Immediate

Format:
cISYS

c
The variable *c* can be one of these values:

s
Displays a short form of the ISYS command

z
Displays data sharing group information for the IMS (if applicable)

If IMS is installed without Remote Site Recovery, the system displays ISYS information as shown in the following example:

```
  ISYS  IMS Version 12.1.0                Subsystem ID = 'IC1C'
+      MVS/ESA -- SP7.1.3                IRLM Release  2.3
+      IMS Restart date = 13.139 (Local) IMS Restart time = 14:38:35 (Local)
+      IMS Restart date = 13.139 (UTC)   IMS Restart time = 18:38:35 (UTC)
+      Checkpoints taken = 1             Current Checkpoint id = 20131391838350
+      MPPs active = 1                   BMPs active = 0
+      Applications scheduled = 0        Transactions queued = 0
+      Msg Enqueue rate =.00/sec         Msg Dequeue rate =.00/sec
+      System Dsn OSAM I/O's = 10        DLS OSAM I/O count  = 0
+      >>> Remote Site Recovery not installed <<<
+      IMS Base LUname = IMSLUC1C        IMS Base Network Id = USCACO01
+      APPC Status = ENABLED             APPC Desired Status = ENABLED
+      RACF Option = NONE                Generic Resources LU = --N/A--
+      LU62 Active Sync Convs = 0        LU62 Active Async Convs = 0
```

Figure 50. ISYS information for IMS without Remote Site Recovery

If you have IMS with Remote Site Recovery installed, the system displays ISYS information that is shown in the following example:

```
 ISYS  IMS Version 10.1.0                 Subsystem ID = 'IA1A'
+      MVS/ESA - SP7.1.2                  IRLM Release 2.2
+      IMS Restart date = 11.066 (Local)  IMS Restart time = 09:24:58 (Local)
+      IMS Restart date = 11.066 (UTC)    IMS Restart time = 14:24:58 (UTC)
+      Checkpoints taken = 1              Current Checkpoint id = 20110661424585
+      MPPs active = 1                    BMPs active = 0
+      Applications scheduled = 62        Transactions queued = 0
+      Msg Enqueue rate =.00/sec          Msg Dequeue rate =.00/sec
+      System Dsn OSAM I/O's = 300        DLS OSAM I/O count  = 0
+      RSR Type = ACTIVE                  Global Service Group = IA1A
+      Service Group = IA1A               Transport Mgr SSID = TA1A
+      Readiness level = -N/A-
+      Transport Manager = ACTIVE         VTAM Connection = ACTIVE
+      IMS Base LUname = N/A              IMS Base Network id = N/A
+      APPC Status = FAILED               APPC Desired Status = ENABLED
+      RACF Option = NONE                 Generic Resources LU = --N/A--
+      LU62 Active Sync Convs = 0         LU62 Active Async Convs = 0
```

Figure 51. ISYS information for IMS with Remote Site Recovery

If IMS with Remote Site Recovery is installed, the system displays sISYS information that is shown in the following example:

```
 sISYS MPPs active = 1                    BMPs active = 0
+      Applications scheduled = 62        Transactions queued = 0
+      Msg Enqueue rate =.00/sec          Msg Dequeue rate =.00/sec
+      System Dsn OSAM I/O's = 300        DLS OSAM I/O count  = 0
+      RSR Type = ACTIVE                  Global Service Group = IA1A
+      Service Group = IA1A               Transport Mgr SSID = TA1A
+      Readiness level = -N/A-
+      Transport Manager = ACTIVE         VTAM Connection = ACTIVE
+      IMS Base LUname = N/A              IMS Base Network id = N/A
+      APPC Status = FAILED               APPC Desired Status = ENABLED
+      RACF Option = NONE                 Generic Resources LU = --N/A--
+      LU62 Active Sync Convs = 0         LU62 Active Async Convs = 0
```

Figure 52. sISYS information for IMS with Remote Site Recovery

The **zISYS** command displays the same output as the **ISYS** command with or with Remote Site Recovery, but offers one more line of data for the data sharing group

All formats of the ISYS command (with a blank, z, or s in column 1) include the following fields:

- Number of active MPPs
- Number of active BMPs
- Number of applications that are scheduled since the last warm start
- Number of transactions in the input queue
- Message enqueue and dequeue rates
- OSAM I/Os issued by the control region (for example, OLDS I/O)
- Number of database OSAM I/Os that complete in the DLS address
- Space
- Remote Site recovery information
- APPC/IMS status summary
- Data sharing group for the IMS system, if applicable (Enter **z** in column 1 of the ISYS command.)

MSYS
Displays general information about the state of the z/OS system.

Type:
Immediate

Format:
cMSYS

c

The variable *c* can be one of these values:

s

Displays a short form of the MSYS command

x

Displays the following more rate information:

- Processor rate for the Control Region and Dependent regions
- System (SIO) rate for the Control Region and Dependent Regions
- Paging rate for the Control Region and Dependent Regions

MSYS accepts an argument of *s* or *x* in its label field. sMSYS gives a short form of the MSYS display.

As shown in the following figure, the MSYS command displays the amount of memory that is used by all system control address spaces and the number of active processors.

```
MSYS  System CPU usage = 65.12%          System SIO rate = 13.45/sec
+      IMS CPU usage = 34.45%             IMS SIO rate = 7.35/sec
+      Average IMS CPU = 33.49%           IMS SIO average = 5.55/sec
+      Number of active CPUs = 1
+                       --Virtual Storage---   ----------Working Set-----------
+                       Below 16M  Above 16M        Real   Expanded  Total WKST
+      Control Region :     1408K     11316K       1492K        N/A       1492K
+      DBRC    Region :      564K      8832K         80K        N/A         80K
+      DLS     Region :      708K      8952K        120K        N/A        120K
+      IRLM    Region :      272K      9016K        296K        N/A        296K
```

Figure 53. Typical z/OS system information display

The fields in the MSYS display are explained in the following list.

System CPU usage
Displays the total processor usage for the entire z/OS system. OMEGAMON for IMS calculates system processor usage over the last SRM interval.

IMS CPU usage
Displays the amount that the IMS control region and its dependents expend, which includes both TCB and SRB times. OMEGAMON for IMS calculates IMS processor usage over the last OMEGAMON for IMS cycle, which is a shorter interval than the SRM interval.

The processor figures are percentages of the CEC, and therefore vary between 0 - 100, rather than between 0 and (100*(number of online processors)).

Average IMS CPU
Displays the average amount of processor that the IMS control region and its dependents expend since this OMEGAMON for IMS session is started.

Number of active CPUs
Displays the number of active processors.

System SIO rate
Displays the SIO rate across all of z/OS.

IMS SIO rate
Displays the SIO rate within all regions.

IMS SIO average
Displays the average SIO rate within all regions since this OMEGAMON for IMS session started.

Virtual Storage
Displays the number of virtual pages IMS has currently GETMAINed. (This number is suitable as an estimate for the size of the REGION= parameter on the z/OS EXEC JCL statement.)

Working Set
Displays real and expanded memory usage, and the current working set size.

Note: In this example, N/A indicates that expanded storage is not available on this processor.

IMS online change analysis

Use the OCHG command to analyze IMS online change operations.

OCHG

Analyzes online change operations.

Type:

Immediate

The following is an example of the OCHG command display.

```
 OCHG  Online change is not in progress
+ Current online change id: 3               Current nucleus suffix: 0
+ Modify work area address: 00000000        Modstat work area address: 00118298
+ MODBLKSA is active       ACBLIBA  is active          FORMATA  is active
+
+                          DATE         TIME
+ LAST MODIFY PREPARE      97157        09:59:068
+ LAST MODIFY COMMIT       97157        09:59:128
+ MODSTAT    VOL=IMS500    DSN=IMS.V500.MODSTAT
+ MODSTAT2   NOT XRF
```

Figure 54. OCHG command display

The first line of the display indicates whether an online change operation is in progress. The IMS **/ MODIFY PREPARE** command starts an online change operation.

The second line of the display shows both the current online change ID and the suffix of the IMS nucleus module (DFSVNUCx) currently in use.

The third line of the display shows the address of the modify work area, which IMS allocates during the PREPARE phase of online change. This work area contains the change, add, and delete lists for every class of resource eligible for online change, and various status and option flags. IMS uses the information that this work area contains during the COMMIT phase of online change. IMS deletes the work area at the end of COMMIT or ABORT phase processing.

The fourth line of the display shows which of the MODBLKSA/MODBLKSB, IMSACBA/IMSACBB, and FORMATA/FORMATB data sets area is in use. IMS uses the online change function to swap between one or more of these pairs of data sets.

The fifth line of the display shows the date and time that the command is issued. If this information is not available, OCHG displays N/A.

The sixth line of the display shows the date and time that the command is issued.

The seventh line of the display shows the MODSTAT data set name and the volser.

The eighth line of the display displays if XRF is available. If it is, the VOLSER and DSN displays for the standby system. If XRF is not available, MODSTAT2 displays NOT XRF.

IMS might suspend MFS I/O during an online change in which FMTLIB is specified. The following list provides description of the possible MFS I/O statuses with an explanation of each line.

```
        I/FETCH IS OK|PENDING
        P/FETCH IS OK|PENDING
        I/FETCH DIR READ IS OK|PENDING
        P/FETCH DIR READ IS OK|PENDING
        MFSTEST I/O is OK|PENDING
```

- The vertical bar between *OK* and *PENDING* indicates that either *OK* or *PENDING* is displayed, but not both.
- I/FETCH describes immediate fetch.
- P/FETCH describes pre-fetch.
- I/FETCH DIR READ describes immediate fetch for the PDS directory.
- P/FETCH DIR READ describes pre-fetch for the PDS directory.

- MFSTEST describes status of I/O to the MFS test PDS.

In addition to MFS library status, the OCHG command also displays if security failed during an online change. Listed as follows are the possible security statuses.

```
Pwd security : password security failed
Term security : terminal security failed
TCMD security : terminal command security failed
Userid security : user id security failed
AGT security : application group security failed
```

IMS data set information

Use the IDDN command to display information and statistics about IMS data sets.

IDDN
Displays information and statistics about IMS data sets.

Type:
Major

Each minor command of IDDN displays the following information:

- The data set name.
- The volume serial number and unit address of the disk on which the data set is stored.
- The logical record length and block size of the data set.
- The total number of I/Os that IMS issued against the data set.
- The I/O rate over the last OMEGAMON for IMS cycle.

Note: The first time you issue IDDN, OMEGAMON for IMS displays “Initializing” for the rates.

ACB
Displays ACBLIB data set information.

Type:
Minor of IDDN

Format:
ACBc

OMEGAMON for IMS displays information about both the A and the B data sets and highlights the active data set of each pair. To specify a data set, type one of the following arguments after the command name.

A
Displays only the A data set.

B
Displays only the B data set.

*****
Displays only the active data set.

For example, ACBA displays information about the A data set only.

The following example shows the **ACB** command display.

```
 IDDN  IMS Dataset Information
 acb   DDNAME = ACBLIBA                 Status = Inactive -- Closed
+      DSNAME = IMS.V500.ACBLIBA
+      Unit address = 14B               Volume = IMS100
+      Logical record length =     0    Blocksize =     0
+      CTL I/O Count =       0          CTL I/O Rate =     .00 per second
+      DLS I/O Count =     176          DLS I/O Rate =     .00 per second
+      ------------------------------------------------------------------------
+      DDNAME = ACBLIBB                 Status = Active -- Open
+      DSNAME = IMS.V500.ACBLIBB
+      Unit address = 14B               Volume = IMS100
+      Logical record length =     0    Blocksize = 23476
+      CTL I/O Count =      47          CTL I/O Rate =     .00 per second
+      DLS I/O Count =     176          DLS I/O Rate =     .00 per second
```

Figure 55. ACB command display

If you use LSO=S, the ACBLIB I/O counts display for both the DLISAS region and for the IMS control region.

DDNM

Displays data set and device I/O statistics for a specific DDName.

Type:

Minor of IDDN

The following example shows the DDNM command display.

```
 IDDN  IMS Dataset Information
 ddnm  DI21PARO
       DDNAME = DI21PARO                Status = Open
       Jobname = IMS510AD               ASID = 103

       DSNAME = IMS.V5R1.DI21PARO
       Index CI Size       0            Data CI Size =  4096
       CI Splits =     0                CA Splits       0
          EXCP Count =        3            EXCP Rate =      .00 per second
          UCB     VOLSER     EXCP Count     EXCP Rate
          ----    ------     ----------     ---------
          04F8    PPSMP3              3           .00
```

Figure 56. DDNM command display

This command displays the I/O rates for all the data sets allocated to the specified DDName. All the devices for each data set are listed with the I/O rates calculated for the EXCPs to a device for the data sets.

LGMG

Displays long message data set information.

Type:

Minor of IDDN

The following example shows a typical **LGMG** command.

```
 IDDN  IMS Dataset Information
 lgmg  DDNAME = LGMSG                   Status = Open
+      DSNAME = IMS.V500.LGMSG
+      Unit address = 161               Volume = IMS500
+      Logical record length =  3100    Blocksize =  6200
+      I/O Count =       4              I/O Rate =     .00 per second
+
+      Dataset utilization:  = .10%, 1 of 33555382
+              Shutdown = 90.47%
```

Figure 57. LGMG command display

The calculation of message queue data set usage also shows the number of blocks in use.

MDBL

Displays information about MODBLKS data sets.

Type:
Minor of IDDN

Format:
MDBLc

OMEGAMON for IMS displays information about both the A and the B data sets and highlights the active data set of each pair. To specify a data set, type one of the following arguments after the command name.

A
Displays only the A data set.

B
Displays only the B data set.

*****
Displays only the active data set.

For example, MDBLA displays information about the A data set only.

Figure 58 on page 104 shows an example of the MDBL command display.

```
 IDDN  IMS Dataset Information
 mdbl  DDNAME = MODBLKSA                Status = Active -- Open
+      DSNAME = IMS.V500.XA.MODBLKSA
+      Unit address = 161               Volume = IMS500
+      Logical record length =     0    Blocksize = 23476
+      I/O Count =       9              I/O Rate =     .00 per second
+      -------------------------------------------------------------------
+      DDNAME = MODBLKSB                Status = Inactive -- Closed
+      DSNAME = IMS.V500.XA.MODBLKSB
+      Unit address = 14B               Volume = IMS100
+      Logical record length =     0    Blocksize =     0
```

Figure 58. MDBL command display

MDST

Displays MODSTAT data set information.

Type:
Minor of IDDN

Figure 59 on page 104 shows the MDST command display.

```
IDDN  IMS Dataset Information
 mdst  DDNAME = MODSTAT                 Status = Open
+      DSNAME = IMS.V500.MODSTAT
+      Unit address = 161               Volume = IMS410
+      Logical record length =    80    Blocksize =    80
+      I/O Count =       2              I/O Rate =     .00 per second
```

Figure 59. MDST command display

MFS

Displays MFS format library data set information.

Type:
Minor of IDDN

Format:
MFSc

OMEGAMON for IMS displays information about both the A and the B data sets and highlights the active data set of each pair. To specify a data set, type one of the following arguments after the command name.

A
Displays only the A data set.

B
: Displays only the B data set.

*****
: Displays only the active data set.

For example, MFSA displays information about the A data set only. The example that follows shows a typical MFS command.

```
 IDDN  IMS Dataset Information
 mfs   DDNAME = FORMATA                 Status = Active -- Open
+      DSNAME = IMS.V500.FORMATA
+      Unit address = 14B               Volume = IMS100
+      Logical record length = 23476    Blocksize = 23476
+      I/O Count =      36              I/O Rate =     .00 per second
+      -------------------------------------------------------------------
+      DDNAME = FORMATB                 Status = Inactive -- Closed
+      DSNAME = IMS.V500.FORMATB
+      Unit address = 14B               Volume = IMS100
+      Logical record length =     0    Blocksize = 23476
```

Figure 60. MFS command display

QBLK

Displays queue blocks data set information.

Type:
: Minor of IDDN

The following example shows a typical **QBLK** command.

```
 IDDN  IMS Dataset Information
 qblk  DDNAME = QBLKS                   Status = Open
+      DSNAME = IMS.V500.QBLKS
+      Unit address = 161               Volume = IMS410
+      Logical record length =    56    Blocksize =  6200
+      I/O Count =      78              I/O Rate =     .00 per second
+
+      Dataset utilization:  = .23%, 27 of 33565882
+              Shutdown = 99.13%
```

Figure 61. QBLK command display

The calculation of message queue data set usage also shows the number of blocks in use.

RCNS

Displays database recovery control (DBRC) RECON data set information.

Type:
: Minor of IDDN

The following example shows a typical RCNS command.

```
 IDDN  IMS Dataset Information
 rcns  DDNAME = RECON1                  Status = Active -- Open
+      DSNAME = IMSV.V500.RECON01
+      Unit Address = 161               Volume = IMS500
+      Index CI Size =  4096            Data CI Size =  4096
+      CI Splits =     0                CA Splits =     0
+      I/O Count =      39              I/O Rate =     .00 per second
+      -------------------------------------------------------------------
+      DDNAME = RECON2                  Status = Active -- Open
+      DSNAME = IMSV.V500.RECON02
+      Unit Address = 161               Volume = IMS500
+      Index CI Size =  4096            Data CI Size =  4096
+      CI Splits =     0                CA Splits =     0
+      I/O Count =      39              I/O Rate =     .00 per second
+      -------------------------------------------------------------------
+      DDNAME = RECON3                  Status = Not Open
```

Figure 62. RCNS command

RDS

Displays restart data set information.

Type:
Minor of IDDN

The following example shows a typical **RDS** command.

```
IDDN  IMS Dataset Information
 rds   DDNAME = RDS                     Status = Open
+      DSNAME = IMS.V500.RDS
+      Unit address = 161               Volume = IMS500
+      Logical record length = 23476    Blocksize = 23476
+      I/O Count =       5              I/O Rate =     .00 per second
```

Figure 63. RDS command display

SHMG

Displays short message data set information.

Type:
Minor of IDDN

The following example shows a typical **SHMG** command.

```
 IDDN  IMS Dataset Information
 shmg  DDNAME = SHMSG                   Status = Open
+      DSNAME = IMS.V500.SHMSG
+      Unit address = 161               Volume = IMS410
+      Logical record length =   312    Blocksize =  6200
+      I/O Count =       9              I/O Rate =     .00 per second
+
+      Dataset utilization:  = .22%, 13 of 33560317
+              Shutdown = 98.32%
```

Figure 64. SHMG command display

The calculation of message queue data set usage also shows the number of blocks in use.

SPA

Displays scratch pad area (SPA) data set information.

Type:
Minor of IDDN

The example that follows shows a typical **SPA** command.

```
IDDN  IMS Dataset Information
  spa   DDNAME = IMSSPA                 Status = Open
 +      DSNAME = IMS.V500.SPA
 +      Unit address = 161              Volume = IMS410
 +      Logical record length =  6000   Blocksize =  6000
 +      I/O Count =       0             I/O Rate =     .00 per second
```

Figure 65. SPA command display

The message **>> No SPA Dataset defined - might use core SPAs only <<** indicates that IMS does not allocate an SPA data set.

SUMM

Displays data set I/O statistics for IMS system data sets as shown in the following figure.

Type:
Minor of IDDN

```
IDDN  IMS Dataset Information
  SUMM
        Jobname  DDName   Status        EXCP Count  EXCP Rate
        -------- -------- ------------  ----------  ---------
        IMS510AC IMSACBA  ACT/OPEN              53        .00
        IMS510AD IMSACBA  ACT/OPEN             213        .00
        IMS510AC IMSACBB  INACT/CLOSED           0        .00
        IMS510AD IMSACBB  INACT/CLOSED           0        .00
        IMS510AC IMSTFMTA ACT/OPEN               0        .00
        IMS510AC IMSTFMTB INACT/CLOSED           0        .00
        IMS510AC LGMSG    OPEN                 129        .00
        IMS510AC SHMSG    OPEN                  58        .00
        IMS510AC QBLKS    OPEN                  66        .00
        IMS510AC FORMATA  ACT/OPEN              48        .00
        IMS510AC FORMATB  INACT/CLOSED           0        .00
        IMS510AB RECON1   OPEN                 237        .00
        IMS510AB RECON2   OPEN                 223        .00
        IMS510AC IMSRDS   OPEN                  13        .00
        IMS510AC MODBLKSA ACT/CLOSED            10        .00
        IMS510AC MODBLKSB INACT/CLOSED           0        .00
        IMS510AC MATRIXA  ACT/CLOSED            12        .00
        IMS510AC MATRIXB  INACT/CLOSED           0        .00
        IMS510AC MODSTAT  CLOSED                 2        .00
```

Figure 66. SUMM command display

This command displays an overview of the I/O rates for the various IMS system data sets.

TMFS

Displays test MFS format library data set information.

Type:

Minor of IDDN

Format:

TMFSc

OMEGAMON for IMS displays information about both the A and the B data sets and highlights the active data set of each pair. To specify a data set, type one of the following arguments after the command name.

A

Displays only the A data set.

B

Displays only the B data set.

*****

Displays only the active data set.

For example, TMFSA displays information about the A data set only.

The following example shows a typical **TMFS** command.

```
IDDN  IMS Dataset Information
  tmfs  DDNAME = IMSTFMTA                Status = Active -- Closed
 +      DSNAME = IMS.TFORMAT
 +      DSNAME = IMS.V500.FORMATA
 +      Unit address = 14B               Volume = IMS100
 +      Logical record length = 23476    Blocksize = 23476
 +      ----------------------------------------------------------------------
 +      DDNAME = IMSTFMTB                Status = Inactive -- Closed
 +      DSNAME = IMS.TFORMAT
 +      DSNAME = IMS.V500.FORMATB
 +      Unit address = 14B               Volume = IMS100
 +      Logical record length =     0    Blocksize =     0
```

Figure 67. TMFS command display

IMS transaction rates

Use the ITX command to display the current IMS transaction rates.

ITX

Displays the current IMS transaction rates.

Type:

Immediate

Format:

cITXnnn

ITX displays three types of information, producing rate displays by transaction name, by OMEGAMON for IMS transaction group name, and by IMS class. For a description of the transaction group concept, refer to "Defining groups" on page 135.

The three-digit argument is used as a display threshold by rate. For example, the command ITX003 displays only rates if they are greater than or equal to three transactions per minute.

The rates ITX displays are an average that it compiles over the previous 60-second interval, so the ITX display changes only once every minute, as each interval completes. This is true for all except the first 60 seconds, when ITX recalculates its display every cycle until it collects the first interval of data. After that first interval, the display freezes and only changes as each succeeding interval completes.

When you run ITX processing for the first time, it produces the message Initializing. After this point, OMEGAMON for IMS collects transaction processing rate data whether ITX is on the screen or not. That is, ITX does not require to reinitialize if you remove the command from the screen. The example that follows shows the transaction names, OMEGAMON for IMS transaction groups, and IMS classes for transactions whose rates are equal to or greater than two transactions per minute. The following figure is an example of an ITX display.

```
 ITX002|Trxname /min 0_2_4_6_8_|Group   /min 0_2_4_6_8_|Class   /min 0_2_4_6_8_|
+      |PART      18 ->. . . . |INQUIRE   32 --->. . . |Class001  32 --->. . . |
+      |DLETINV   32 --->. . . |UPDATE    18 ->. . . . |Class004  18 ->. . . . |
```

Figure 68. ITX example

If you enter R in its label field, ITX resets the IMS class information to reflect /ASSIGN changes.

Fast path transaction rates

Use the FTX command to display fast path transaction rates.

FTX

Displays the current rates for fast path exclusive transactions and those fast path potential transactions that run in fast path regions.

Type:

Immediate

Format:

cFTXnnn

FTX does not show transactions that run in IMS regions. It displays these rates by balancing group (BALG), not by CLASS or GROUP, as does ITX.

The three-digit argument is uses as a display threshold by rate. For example, the command FTX001 displays only rates if they are greater than or equal to one transaction per minute.

The rates FTX displays are an average that is compiled over the previous 60-second interval, so the FTX display changes only once every minute, as each interval completes. This is true for all except the first 60 seconds, when FTX recalculates its display every cycle until it collects the first interval of data. After that first interval, the display freezes and only changes as each succeeding interval completes.

When you run FTX for the first time, it produces the Initializing message. After this point, OMEGAMON for IMS collects transaction rate data whether FTX is on the screen or not. That is, FTX does not require to reinitialize if you remove the command from the screen. The following example shows the results of the FTX command.

```
FTX030|BALG FPM/MIN 0       100       200       300       400       500  |
+      |DBFSAMP3 183 ------------------> .    .    .    .    .    .    .   |
+      |DBFSAMP2  45 ----->   .    .    .    .    .    .    .    .    .   |
+      |DBFSAMP4 246 ----------------------->.    .    .    .    .    .   |
+      |DBFSAMP6 426 ------------------------------------------> .    .   |
==========================================================================
1FTX030|BALG FPM/MIN 0        20        40        60        80       100  |
+      |DBFSAMP3 183 ++++++++++++++++++++++++++++++++++++++++++++++++++>  |
+      |DBFSAMP2  45 ----------------------> .    .    .    .    .    .   |
+      |DBFSAMP4 246 ++++++++++++++++++++++++++++++++++++++++++++++++++>  |
+      |DBFSAMP6 426 ++++++++++++++++++++++++++++++++++++++++++++++++++>  |
```

Figure 69. FTX command display

FTX accepts an alphanumeric argument in its label field. FTX uses this argument as a scaling factor to determine the maximum transaction rate for the graph. If you do not supply a scaling factor, FTX uses five as its default, which means that the graph plots from 0–500 transactions per minute.

FTX accepts the following scaling factors:

1
: 0–100

2
: 0–200

3
: 0–300

4
: 0–400®

5
: 0–500

6
: 0–600

7
: 0–700

8
: 0–800

9
: 0––900

A
: 0–1500

B
: 0–2000

C
: 0–2500

D
: 0–3000

E
: 0–3500

F
: 0–4000

IMS trace table entries

Use the TRAC command to display IMS trace table entries.

TRAC
Displays IMS trace table entries.

Type:
Immediate

Format:
TRACcc PST=n FUNC=xx

cc
Specifies the type of trace:

DG
Disk trace log

DL
DL/I trace table entries

DS
Dispatcher trace table entries

FO
Force trace

FP
Fast path trace

LA
Latch trace

SC
Scheduler trace table entries

SS
Subsystem trace

PST=n
Specifies the number of the PST.

FUNC=xx
Specifies the number of hexadecimal trace entries.

The following example shows a small portion of the dispatcher trace table.

```
TRACDS
+ DS TRACE TABLE
+ BEGINNING ADDR = 00A8F0A0  NEXT ENTRY ADDR = 00A8FCE0
+ 00A8F0A0 19051832 0013B050 40C4D3F4 FF575CB4  IPC ENQ
+ 00A8F0B0 1A051833 00000000 00250025 00000025  IPC RESUME
+ 00A8F0C0 06FE1834 007D7880 50ABA866 00AB53D4  IPOST(ECB=)
+ 00A8F0D0 19021835 00AB53D4 40E3D9C1 FF575DB4  IPC ENQ
+ 00A8F0E0 1A021836 007D7880 00250025 00000025  IPC RESUME
+ 00A8F0F0 05051837 0013B050 40C4D3F4 40000000  RE-DISPATCH
+ 00A8F100 23051838 0013B050 00000000 40A82692  ISERWAIT
     .        .        .        .        .         .
     .        .        .        .        .         .
     .        .        .        .        .         .
```

Figure 70. Dispatcher trace table sample

Time-controlled operations information

You can use Time-Controlled Operations (TCO) to start BMPs at predetermined times and automate image copies and reorganization of databases. You can also issue any IMS command, which includes commands that shut down IMS automatically.

TCO can also be used with the automated operator interface (AOI) to intercept commands or messages for automated operations.

TCO receives all information from a script library. The members in this library contain the IMS commands and time schedule requests. The members can also contain message sets.

A time schedule request in a script can specify:

- A time of day (by hour and minute)
- A time interval
- A specified delay after IMS startup

At the time that is requested in the schedule, TCO can complete the following tasks:

- Load a new script
- Enter commands
- Send message switches
- Send transaction messages

TCOS

Displays the status of Time-Controlled Operations (TCO).

Type:

Immediate

TCOS also displays the current script member name and whether TCO is active or inactive.

TCOC

Displays the contents of the current TCO script member.

Type:

Immediate

Format:

TCOCcc

Where *cc* can be:

blank

Displays all commands and messages in the current script member.

ST

Displays startup commands and messages only.

DL

Displays delayed startup commands and messages only.

TD

Displays time-of-day commands and messages only.

To display all the commands and messages in the current script member, enter TCOC without any argument, as illustrated in the following figure.

```
TCOC
+ IMS Startup commands/messages:
+ ELEM# TIME    USER EXIT   COMMAND/MESSAGE
+ ----- ------  ---------   ---------------
+     1 START   DFSTXIT0    /STA DC
+     1 START   DFSTXIT0    /STA LINE 1 PTERM 1
+
+ DELAYED STARTUP COMMANDS/MESSAGES:
+ ELEM# TIME    USER EXIT   COMMAND/MESSAGE
+ ----- ------  ---------   ---------------
+     1 11:55   USEREXT1    /STA PGM ALL
+     1 11:55   DFSTXIT0    /STA TRAN ALL
+
+ TIME-OF-DAY COMMANDS/MESSAGES
+ ELEM# TIME    USER EXIT   COMMAND/MESSAGE
+ ----- ------  ---------   ---------------
+     1 12:00   DFSTXIT0    /DIS A
+     2 13:00   DFSTXIT0    /DIS A
+     3 14:00   DFSTXIT0    /DIS A
+     4 15:00   DFSTXIT0    /DIS A
+     5 16:00   DFSTXIT0    /DIS A
```

Figure 71. TCOC command display

System information graphs

Use the PLOT command to produce graphs of various system resources. OMEGAMON for IMS collects data about IMS resources and stores this data in an internal table. The PLOT command displays information from this internal table.

PLOT

Graphically displays historical information about IMS resources.

Type:

Immediate

Format:

cPLOT aaaa

The PLOT command accepts one, two, or three four-character resource names (*aaaa*) as parameters. If you supply more than three names, the PLOT command displays the message **EXTRANEOUS NAMES IGNORED. PLOT CAN ONLY SHOW THREE PLOTS PER COMMAND**.

The PLOT command displays information about the following resources:

ARVL

Transaction arrival rate (includes fast path if installed).

CPUC

Amount of processor used.

DEQU

Transaction dequeue rate (includes fast path if installed).

INQL

Transaction input queue size (includes fast path if installed).

IORT

IMS I/O rate per second.

PGAV

Private page-in rate per second for all non-dependent regions.

PGCR

Private page-in rate per second for the control region.

PGDB

Private page-in rate per second for the DBRC region.

PGDL

Private page-in rate per second for the DLI region.

PGIR
Private page-in rate per second for the IRLM region.

ROAV
Average region occupancy for all dependent regions.

If you enter a name that the PLOT command does not recognize, it displays the message **NOT DEFINED**.

The PLOT command uses the graph character that you provide in the label field (*c*) to plot each point on the graph. If you do not supply a graph character, PLOT uses the default (an asterisk).

The PLOT command displays information as a vertical bar graph. The following figure shows the results of a typical PLOT command. The vertical bars farthest to the right of the graph represent the most current data. As you move to the left across the graph, the vertical bars represent an averaged value, which the product updates every 2, 4, 8, 16, 32, 64, and 128 cycles.

```
PLOT CPUC IORT
+
+ CPUC  CPU UTILIZATION     IORT  IMS I/Os PER SEC
+  80%  +                   240   +             *
+      =>----------------        =>-------------*--
+  60%  +                   180   +         * * *
+       |                         |         * * *
+  40%  +                   120   +         * * *
+      =>----------------         |       * * * *
+  20%  +                    60  =>-----*-*-*-*-*--
+       | * * * *     * *         | * * * * * * *
+       + - + - + - + - +         + - + - + - + - +
+       Past          Now         Past          Now
+                                       *****
```

Figure 72. PLOT command display

Each graph contains an upper and lower threshold line. When a resource equals or exceeds the upper threshold, that resource is considered in danger. If a resource is below the lower threshold, that resource might or might not be in danger, depending on the resource and your operating environment. Use the PSETmajor command and its minor commands to adjust the upper and lower thresholds.

The column to the left of the current cycle represents data averaged over the last two cycles. If the value in this column exceeds the upper threshold, the PLOT command displays a horizontal line of five highlighted asterisks below the resource graph.

In extended color mode, the PLOT command displays the body of the graph in reverse video. The upper threshold displays in red. The lower threshold displays in blue. The column to the left of the current cycle represents data averaged over the last two cycles. If the value in this column exceeds the upper threshold, the PLOT command displays the body of the graph in red. If the value in the column that represents data averaged over the last two cycles is between the two thresholds, the PLOT command displays the body of the graph in yellow. If the value in the column that represents data averaged over the last two cycles is less than the lower threshold, PLOT displays the body of the graph in blue.

If you enter the command PLOTX instead of the PLOT command (in extended color mode only), each vertical column displays in its own color: red for values above the red line, yellow for values between the two thresholds, and blue for values below the blue line.

PSET
Sets the thresholds that the PLOT command uses dynamically.

Type:
Major

Format:

```
PSET
{aaaa M=n U=n L=n | aaaa}
```

aaaa
: Any of the resource names that the PLOT command can display. The resource name is entered as a minor command to PSET. If you specify the resource name with no other options, the Y-axis and threshold values for the resource are displayed.

M=
: The maximum of Y-axis scale value: the minimum scale value is 8

U=
: The value of the upper threshold

L=
: The value of the lower threshold

The following example shows a PSET command with the ARVL minor command.

```
PSET
ARVL  M=30  U=22  L=7
```

The previous commands set the maximum Y-axis scale value of the transaction arrival rate plot to 30, the upper threshold to 22, and the lower threshold to seven.

The result of the PLOT command for ARVL is shown in the following figure:

```
 PLOT ARVL
+
+ ARVL  MSG ENQ RATE/SEC
+  30   +
+       |
+  22  =>----------------
+       |
+       +             * *
+      =>-*-*-*-*-*-*-*-*
+   7   + * * * * * * * *
+       | * * * * * * * *
+       + - + - + - + - +
+       Past          Now
+
```

Figure 73. PLOT command for ARVL display

The upper threshold value must be greater than the lower threshold value. The maximum Y-axis scale value must be greater than the upper and lower threshold values.

If the lower threshold is greater than the upper threshold or the maximum Y-axis scale value is less than the upper threshold, the PSET command displays an error message and does not set these values.

If you issue one of the PSET minor commands without an M=, U=, or L= parameter, the minor command displays the current M=, U=, and L= parameters for that resource.

Chapter 6. Device information

OMEGAMON for IMS provides commands that display information about disk and tape devices.

Device listing commands

You can use the DLST, GLST, and TLST immediate commands to display device information.

DLST
: Lists all online and offline disks.

  Type:
  Immediate

  A hyphen (-) in the command display indicates the selected offline disks.

GLST
: Lists esoteric and generic device names by device class.

  Type:
  Immediate

  Format:
  GLST <cccc|cccc ...>

  The GLST display can be limited to specific device classes with the following operands:

  b
  Displays all defined device names. Blank is the default.

  ALL
  Displays all device names. ALL produces the same display as blank.

  CHAR
  Displays character reader devices.

  COMM
  Displays communication devices.

  CTC
  Displays channel to channel devices.

  DASD
  Displays direct access storage devices.

  DISP
  Displays display devices.

  TAPE
  Displays tape devices.

  UREC
  Displays unit record devices.

  These operands can be used in combination. For example:

```
GLST TAPE COMM
```

  Displays all tape devices and communications devices, as shown in the following figure:

```
 GLST
+  TAPE DEVICES:
+      3480X    3400-6   TAPE     CART     T3480    3480     3400-9   3400-5
+      3400-3
+
+  COMMUNICATIONS DEVICES:
+      3705
```

Figure 74. Tape and communications devices

Use of the optional operands is limited to z/OS 4.1 and higher versions. See also the GDEV command.

TLST

Selects all tape drives (online/offline).

Type:
Immediate

A hyphen (-) in the command display indicates the offline tape drives selected.

Disk information

You can issue disk information major commands to select disks and minor commands to display detailed information about the disks.

Disk select major commands

Use the following major commands to select one or more disks. After you select the disks, use the minor commands to display detailed information about the disks.

DEV

Selects a disk with volser *cccccc* or address *xxx* or *xxxx*.

Type:
Major

Format:
DEV cccccc|xxxx

Note: The DEV command accepts input in either a 3-digit or 4-digit format.

If you supply the volser or the address of a disk, the DEV command displays the volser, the address, and the online or offline status of the disk.

If you enter:

```
      DEV 5200
```

The result is as follows:

```
      DEV 5200 volser=TSO099  Online  Alloc
```

Alternatively, you can provide the volser (TSO099).

DEVL

Selects list of online disks by volser *cccccc* or unit address *xxxx*.

Type:
Major

Format:
DEVL cccccc ... cccccc|xxxx ... xxxx

The following example shows disks at addresses 123 and 141 and volsers TSO021 and TSO022:

```
      DEVL 1230 TSO021 TSO022 1410
```

If you list an invalid address or volser, or if you specify the same disk twice, OMEGAMON for IMS eliminates the invalid or duplicate value from the list.

DEVP
Selects a list of online disks by using patterns that are set with the `.SPT` command.

Type:
Major

Format:
DEVP/n

The variable *n* is a number from 0 to 9 indicating the pattern that is set with the `.SPT` immediate command.

For example, you can set the pattern and then display all online disks that begin with TSO as shown in this figure:

```
.SPT/9   TSO*
DEVP/9   TSO024   TSO025   TSO021   TSO022   TSO023   TSO069
```

Figure 75. Online disks display

DISK
Selects online disks.

Type:
Major

An argument of AL (DISKAL) displays all offline disks and those disks that are online.

DSKB
Selects busy disks.

Type:
Major

DSKC
Selects disks with suspended channel programs.

Type:
Major

DSKE
Selects permanently resident disks.

Type:
Major

DSKG
Selects mass storage (MSS) virtual disks.

Type:
Major

DSKM
Selects disks that are waiting on mounts.

Type:
Major

DSKN
Selects disks with volsers that start with *cc*.

Type:
Major

Format:
DSKNcc

The following example selects disks with volsers that begin with the characters TS, such as all TSO disks:

```
DSKNTS
```

DSKP
Selects DASD volumes with a mount status of PUBLIC.

Type:
Major

DSKQ
Selects disks with I/O queue length of *nn* or more.

Type:
Major

Format:
DSKQnn

The DPLTnn minor command plots a microscopic analysis of device usage. You can use it to investigate disks with I/O queues.

DSKR
Selects disks with a RESERVE currently issued from this processor.

Type:
Major

DSKS
Selects DASD volumes with a mount status of STORAGE.

Type:
Major

DSKU
Selects disks with UCBnames starting with *xx* or *xxx*.

Type:
Major

Format:
DSKUxx

Note: DSKU command now accepts input in either a 2-digit or 3-digit format.

The variable *xx* or *xxx* specifies the UCBnames with which the disks begin. This might or might not correspond to control unit *xx* or *xxx*. DSKUxx does not consider alternate paths.

The following example selects disks with UCBnames that start with 58, which includes disks 580 through 58F:

```
DSKU58
```

DSKV
Selects DASD volumes with a mount status of PRIVATE.

Type:
Major

GDEV
Lists devices with name *cccccc*.

Type:
Major

Format:
nGDEV cccccc

This major command lists disk and tape devices with the generic name *cccccc*. The value of *n* can be one of the following values: the less-than symbol (<) to display all devices, or the numbers 1 to 9 and the letters A to Z (representing 10 - 35) to display individual rows of the list of devices.

For example, SYSDA is a generic device name in the system. To list all devices with the generic name SYSDA, enter the following keyword:

```
<GDEV SYSDA
```

You can specify the number of entries in the device name table with the GDEVUCBS keyword of the **.SET** command. The maximum is 4000.

Note: Use the GLST major command to list the generic names in the system.

3340
Displays 3340 disks.

Type:
Major

3350
Displays 3350 disks.

Type:
Major

Format
3350c

An argument of *P* displays the disks that are attached to 3880-11 or 3880-21 buffered-paging facility control units. In addition, for non-base exposures, the unit address and exposure number display instead of the volser.

3375
Displays 3375 disks.

Type:
Major

3380
Displays 3380 disks.

Type:
Major

3390
Displays 3390 disks.

Type:
Major

Selected disk minor commands

After you issue a disk major command, you can use the following minor commands to display information about the disk drives selected by the disk major command.

DADR
Displays the unit address of the device.

Type:
Minor of disk majors

DALC
Displays number of allocations to the device.

Type:
Minor of disk majors

DCAT

Displays whether a device is static, installation-static, or dynamic.

Type:
Minor of device majors, for example, DISK.

DCAT displays **STATIC**, **I-STATIC**, or **DYNAMIC**, to indicate the device category.

The DCAT command applies to z/OS 4.2 and higher versions.

DIO

Displays EXCPs issued to a device.

Type:
Minor of disk majors

DIO requires RMF.

Note: If RMF is not currently monitoring the device when you issue DIO, the command displays the message UNMNITRD.

DIOQ

Displays I/O queue length on the disk.

Type:
Minor of disk majors

DOPN

Displays number of open DCBs and ACBs on the device.

Type:
Minor of disk majors

DPIN

Displays whether a device is pinned.

Type:
Minor of device majors, for example, DISK.

DPIN displays **PINNED**, **UNKNOWN**, or a field of blanks, to indicate the device category.

The DPIN command applies to z/OS 4.2 and higher versions.

DPLT

Displays device activity every *nn* milliseconds.

Type:
Minor of disk majors

Format:
DPLTnn

The DPLT command allows inspection of processes that occur between OMEGAMON for IMS cycles. For any major device command, the DPLT command takes 50 samples every *nn* milliseconds. The DPLT command plots only the first device that the major specifies. Each sample indicates changes in device status and user.

As the plot progresses from left to right, a number of fields show either the status of the address space or its activity since the last sample. One column in the display represents each sample.

If you use the DPLT command as a minor command of the DSKQ command, OMEGAMON for IMS samples only when there is a need, such as when a disk has an I/O queue length of *nn* or more and is, therefore, selected by the DSKQ command.

To monitor a specific device, use the DPLT command as a minor command of DEV *xxx*, where *xxx* specifies the device address.

The following figure shows an example of the DPLT command that is used with the DSKQ major command.

```
 DSKQ01 z/OS005
 DPLT03 _____ OMEGAMON Peek at unit=138 volser=z/OS005 __________
+      |DBsy  DDDD     DD  DDDDDDD   DDDDDDDDDDDDDDDDDDD   DDDD |    Samples: 50
+      |CBsy  C    CCCC                                         |   Interval: 4
+      |Chan              HH                                    |      I/O's: 6
+      |IOQ   1    11221111          11223322222222111111  2222 |
+      |I/O# >33 >34    >35         >36   >37     >38      >39  |
+      |CPU   |||..|..|...||.......|||......|......|||......||..|
+      |User >PAYROLL   >SORT       >PAYROLL      >SORT         |
+      |Cyl  >280>281   >403        >282  >283    >404     >284 |
+      |Nrdy                                                    |
+      |Resv                                                    |
```

Figure 76. DPLT command with DSKO

To report on the I/O# subfield properly, the DPLT command requires RMF to be monitoring the device.

The Interval value to the right is the true interval between samples, which are calculated after the DPLT command completes processing. On a system that runs perfectly, this number is the same as the *nn* sampling interval. This number might vary because of your system workload.

The fields in the DPLT display include the following.

DBsy
: *D* indicates device busy.

  S (XA and ESA) indicates suspended channel program.

IOQ
: Length of IOQ. For example, a value greater than nine but less than 36 is given by a letter of the alphabet, where A=10, B=11. A value greater than 35 is given by a plus (+) sign.

I/O#
: Wraparound I/O#. (The > indicates the point at which OMEGAMON for IMS calculates a new wraparound I/O#. The new number displays to the right of the > and indicates the end of the I/O.

CPU
: Can be one of these symbols:

  |
  : CPU dispatchable. This character indicates that the address space has at least one TCB ready to be dispatched.

  .
  : Waiting. This character indicates that no TCBs are ready to run.

User
: Current user. (The > indicates the point at which a new user had the device.)

  Note: OMEGAMON for IMS might not be able to identify the user of a device if that user is a system routine or utility that issues its own seek commands.

Cyl
: Cylinder address. (The > indicates the point at which a new cylinder is accessed.) Rls in this field indicates that a stand-alone release is in progress, and therefore no cylinder is involved.

Nrdy
: Not ready.

Resv
: Device reserved this CPU.

Figure 76 on page 121 shows that the DSKQ01 command selected a disk with a volser of z/OS005 because it had an I/O queue length of one or more. The minor command DPLT03 plots the activity on the selected device at three-millisecond intervals (OMEGAMON for IMS always takes 50 samples). The resulting plot shows two jobs (PAYROLL and SORT) competing for the disk arm at cylinder addresses 280 and 403. This explains the I/O queue on the selected device. Even though a three-millisecond interval (DPLT03) is specified, the actual interval that displays is 4 milliseconds (Interval:

4). This happens because other address spaces operated at a higher priority than OMEGAMON for IMS at the time.

DRES
Displays device reserve count from this processor.

Type:
Minor of disk majors

DSTA
Displays mount status.

Type:
Minor of disk majors

Status can be:

- PRIVATE
- PUBLIC
- STORAGE

DTYP
Displays disk type.

Type:
Minor of disk majors

Disk type can be 3380, 3390, and so on.

DUSR
Displays current user of device.

Type:
Minor of disk majors

DVMP
Displays unit control block (UCB) hex memory dump.

Type:
Minor of disk majors

The DVMP command dumps the UCB, the UCB prefix, and all appropriate extensions for the disk. It also shows the device status.

The following screen shows a typical DVMP display.

```
DEV      SYS640
 dvmp  Mount Status: Perm_Res Private
+      Status:
+      User: *MASTER*            Waiting I/O's:
+      Status: Ch_Active - Suspended Channel Program <Paging Device>
+
+       UCB Prefix: 001188
+      00000000 00FF39B4
+
+       UCB Common + DASD Device Dependent Segment: 001190
+      008BFF8C 0240A201 B0000100 00F1F4F0 3010200E 00008D38 19D70100 E2E8E2F0
+      F2F45000 00000400
+
+       Common Extension: 008D38
+      00000000 18820040 000A0000 F755001A 01000000 00022898 0000EAF0 00100005
```

Figure 77. DVMP display

DVOL
Displays the volser of a selected device.

Type:
Minor of DEV and DEVL

ICHPn

Displays installed channel paths for the disk (XA).

Type:
Minor of disk majors

The value of n is a number from 1 to 8.

OCHP

Displays online channel paths for the disk (XA or ESA).

Type:
Minor of disk majors

Format:
OCHPn

The value of *n* is a number from 1 to 8.

Examples of disk commands

The following information shows examples that use the disk commands.

The following example shows you how to use the DSKU major command to monitor the I/O rates of devices at 160 to 16F.

```
 DSKU16  IMS001    IMS002  ....
 dadr       160       161  ....
 dio .R     1.4         0  ....
================================================================================
 DSKU16  SCRAT1    SCRAT2  ....
 dadr       168       169  ....
 dio .R     2.5       8.7  ....
```

Figure 78. DSKU command

The next example shows I/O queue length. For any device with an I/O queue length greater than or equal to 2, the minor commands display the current user, the I/O queue length, and the address.

```
 DSKQ02   z/OSRES   SPOOL
 dusr   *MASTER*    JES2
 dioq          3       2
 dadr        163     167
```

Figure 79. I/O queue length

The next example selects all private disks.

```
 DSKV     DB0021   DB0022   DB0023   DB0024   KN0056   KB0087
 dusr   MESSAGE1                            MESSAGE1
 dio ..R      2.3                                 5.3
```

Figure 80. Private disks

The next example selects all disks whose volsers begin with WO.

```
 DSKNWO  WORK01  WORK02
 dadr       140     147
```

Figure 81. Disks with WO

Tape information

You can issue tape information major commands to select tape drives and minor commands to display detailed information about the tape drives.

Tape select major commands

Use the following major commands to select one or more tape drives. After you select the tape drives, use the minor commands to display detailed information about the tape drives.

DEV

Selects tape drive with volser *cccccc* or address *xxx* or *xxxx*.

Note: DEV Command now accepts input in either a three-digit or four-digit format.

Type:
Major

Format:
DEV cccccc|xxxx

GDEV

Lists devices with name *cccccc*.

Type:
Major

Format:
nGDEV cccccc|xxx

This major command lists disk and tape devices with the generic name *cccccc*. The value of *n* can be one of the following values: the less-than symbol (<) to display all devices, or the numbers 1 - 9 and the letters A to Z (representing 10 - 35) to display individual rows of the list of devices.

For example, TAPEA is a generic device name in the system. To list all devices with the generic name TAPEA, enter the following keyword:

```
<GDEV TAPEA
```

You can specify the number of entries in the device name table with the GDEVUCBS keyword of the **.SET** command. The maximum is 4000.

Note: Use the GLST major command to list the generic names in the system.

TAPE

Selects all online tape drives.

Type:
Major

TPAL

Selects allocated tape drives.

Type:
Major

TPBS

Selects busy tape drives.

Type:
Major

TPCU

Selects all online tapes on the specified control unit.

Type:
Major

Format:
TPCUxx

The value *xx* is a hexadecimal argument.

The following example selects drives 580 through 58F:

```
TPCU58
```

TPFR
Selects online and free tape drives.

Type:
Major

TPFR is the complement of TPAL, which selects tape drives that are online and allocated.

TPMT
Selects drives that are awaiting mounts.

Type:
Major

TPOF
Selects all offline tape drives.

Type:
Major

TP16
Selects 1600 BPI tape drives.

Type:
Major

TP38
Selects 38000 BPI tape drives.

Type:
Major

The DVMP tape minor command dumps the tape class extension for 3480s in both native and compatibility modes.

Note: 3480 tape cartridge drives are also known as 3420-9 in compatibility mode.

TP62
Selects 6250 BPI tape drives.

Type:
Major

TP7T
Selects seven-track tape drives.

Type:
Major

TP80
Selects 800 BPI tape drives.

Type:
Major

Selected tape minor commands

After you issue a tape major command, issue the following minor commands to display information about the tape drives selected by the tape major command.

DCAT

Displays whether a device is static, installation-static, or dynamic.

Type:
Minor of device majors, for example, DISK.

DCAT displays **STATIC**, **I-STATIC**, or **DYNAMIC**, to indicate the device category.

DIO

Displays EXCPs issued to a device.

Type:
Minor of tape majors

DIO requires RMF.

Note: If RMF is not currently monitoring the device when you issue DIO, the command displays the message UNMNITRD.

DPIN

Displays whether a device is pinned.

Type:
Minor of device majors, for example, DISK.\

DPIN displays **PINNED**, **UNKNOWN**, or a field of blanks, to indicate the device category.

DPLT

Displays plot device activity every *nn* milliseconds.

Type:
Minor of tape majors

Format:
DPLTnn

DPLT allows inspection of processes that occur between OMEGAMON for IMS cycles. For any major device command, DPLT takes 50 samples every *nn* milliseconds. DPLT only plots the first device, which the major specifies. Each sample indicates changes in device status and user.

To report on the I/O# subfield properly, DPLT requires RMF to be monitoring the device.

The following screen display shows the DPLT minor command used with the DEV major.

```
 DEV 522 Online
 dplt02 ________ OMEGAMON Peek at unit=522 volser=...... _______
+      |DBsy                                                  | Samples:   50
+      |IOQ   11111111111111111111111111111111111111111111111111| Interval:   6
+      |I/O#                                                  | I/O's:
+      |CPU                                                   |
+      |User                                                  |
+      |Cyl                                                   |
+      |Nrdy  NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN|
+      |Resv                                                  |
```

Figure 82. DPLT minor command used with DEV major

The fields in the DPLT display include the following.

DBsy

D indicates device busy.

S indicates suspended channel program.

IOQ
: Length of IOQ. For example, a value greater than nine but less than 36 is given by a letter of the alphabet, where A=10, B=11. A value greater than 35 is given by a plus (+) sign.

I/O#
: Wraparound I/O#.

CPU
: Can be one of these symbols:

  |
  : CPU dispatchable. This character indicates that the address space has at least one TCB ready to be dispatched.

  .
  : Waiting. This character indicates that no TCBs are ready to run.

User
: Current user.

Cyl
: Cylinder address. R1 in this field indicates that a stand-alone release is in progress, and therefore no cylinder is involved.

Nrdy
: Not ready.

Resv
: Device reserved this processor.

DTYP
Displays device type.

Type:
Minor of tape majors

DUSR
Displays user of device.

Type:
Minor of tape majors

DVMP
Displays UCB hex memory dump.

Type:
Minor of tape majors

For example, the following command dumps the UCB of the device at address 160:

```
DEV 160
DVMP
```

DVMP dumps the tape class extension for 3480s in both native and compatibility modes. The device status indicates the ASSIGN status for 3480s (in native mode) as:

<N-Assign>
Not assigned to any processor.

<S-Assign>
Assigned to a single processor (the drive on which OMEGAMON for IMS is running).

<M-Assign>
Assigned to multiple processors.

Figure 83 on page 128 shows an example of the DVMP command.

```
 TP38       370      371
 dvmp  Mount Status:
+      Status:
+      User: -none-              Waiting I/O's:
+      Status: <N-Assign>
+
+       UCB Prefix: 002008
+      00000000 00000000
+
+       UCB Common + Tape Device Dependent Segment: 002010
+      0010FF00 03700002 00000300 04F3F7F0 33008003 000097E8 00000000 00000000
+      00000000 00000000 00000000 0000E9B0
+
+       Common Extension: 0097E8
+      091C8000 180C0000 01660000 00000000 00000000 00022AF0 0000F0D0 00000000
+
+       Tape Extension: 00E9B0
+      00000000 00000000 00000000 00000000
+
+       Tape Class Extension: 00F0D0
+      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+      00000000 00000000 00000000
```

Figure 83. DVMP command

DVOL

Displays volser of selected device.

Type:
Minor of DEV

If the drive is not allocated, OMEGAMON for IMS displays **--------**.

If a mount is outstanding for the drive, **M*vvvvvv** displays. The variable *vvvvvv* is the volume serial number.

TERP

Displays the number of permanent read/write errors that are accumulated for a tape drive.

Type:
Minor of tape majors

TERT

Displays the number of temporary read/write errors that are accumulated for a tape drive.

Type:
Minor of tape majors

TLBL

Displays type of label (SL, NL, NON-STD).

Type:
Minor of tape majors

BLP tapes display as NL.

TSEQ

Displays the file number that is accessed on an open tape.

Type:
Minor of tape majors

The following figure shows an example that uses the TAPE command with several minor commands.

```
TAPE        380      385
dusr     IMSPROD    BMP01
dvol      001984   009081
dio .R       6.5      5.1
tert           2
terp
```

Figure 84. TAPE command example

Long-term device usage

Use the IDEV major command and its corresponding minor commands to display information about long-term device usage. Long-term usage displays are available if RMF is active for the required devices. If RMF is not active for devices, the average queueing and response time information does not display.

IDEV

Displays device activity data from all IMS system control address spaces.

Type:

Major

Format:

cIDEV

IDEV can also display device activity data from IMS dependent regions. Use region commands (RGNc) to select a region for IDEV. The IDEV minor commands (PDSK, SDSK, and XDSK) display device activity about that region. The IDEV command can use the following label field arguments to select target IMS address spaces or OLDS and WADS devices.

(blank)

Displays devices that are allocated to the IMS control region.

D or S

Displays devices that are allocated to the DLISAS address space.

F or I

Displays devices that are allocated to the IMS control region (default).

L or M

Displays devices that are allocated to the IRLM address space.

O

Displays devices that are used for DASD logging OLDs.

P

Displays devices that are allocated to the IMS dependent region that the preceding RGNc major command selected.

R or C

Displays devices that are allocated to the DBRC address space.

W

Displays devices in use by IMS logging WADs.

PDSK *aaaaaa* selects any device (DASD) whose volume serial number matches the pattern *aaaaaa* specifies.

A pattern consists of alphanumerics and asterisks (*), which act as wildcard characters. For example, the pattern IMS**1 causes OMEGAMON for IMS to select volsers IMS001, IMS011, IMS201.

If an asterisk is the last character in the pattern, any number of characters after that point can match. For example, the pattern VS* selects the volumes VSRESA, VSAM01, VS.

SDSKxx selects any device of the indicated type, which belongs to the *xx* string of devices (that is, any device whose device address begins with *xx*).

XDSK selects any DASD device that exceeds certain thresholds.

These threshold commands do not produce output; these commands control which devices XDSK display.

AVQ

Sets threshold for average IOS queue depth.

Type:

Minor of IDEV

Format:

AVQnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

CON
Sets the threshold for the average device connect time.

Type:
Minor of IDEV

Format:
CONnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

DSC
Sets the threshold for the average device disconnect time.

Type:
Minor of IDEV

Format:
DSCnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

DUT
Sets the threshold for the average device usage.

Type:
Minor of IDEV

Format:
DUTnn|OF

blank
Displays current threshold.

nn
Specifies the threshold for the XDSK command as a percent.

OF
Turns off the previously set threshold.

IOS
Sets the threshold for the IOS queue time.

Type:
Minor of IDEV

Format:
IOSnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

PND
Sets the threshold for the average device pending time.

Type:
Minor of IDEV

Format:
PNDnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

RSP
Sets the threshold for the total device pending time.

Type:
Minor of IDEV

Format:
RSPnnn|OF

blank
Displays current threshold.

nnn
Specifies the threshold for the XDSK command in milliseconds.

OF
Turns off the previously set threshold.

The example that follows shows long-term device usage for volsers whose names match the pattern DI*.

```
DIDEV   Interval Start Time: 13:50:00      Elapsed:  34:13 MN
 PDSK   DI*     <=== Volser pattern
+   Unit Volser  I/O per second  Util% Avg.Q   Resp = IOSQ + Pend + Conn + Disc
+   ---- ------   Total    IMS   ----- ----- ------ ------  -----   ----   ----
+    150 DISK01    27.3    10.3   25.9  1.05   32.1    7.3    4.8   14.5    5.5
+    483 DISK02    10.5     5.7   12.4  0.06   25.1    0.5    4.6   18.1    1.9
```

Figure 85. PDSK under IDEV display

Legend

Volser
Device volume serial number.

Rate
I/O rate (I/Os per second). (total system + IMS)

Util%
Device usage in percent.

Avg.Q
Average IOS queue length.

Resp
Average total device response time in milliseconds.

IOSQ
Average IOS queueing time in milliseconds.

Pend
Average pending time in milliseconds.

Conn
Average connect time in milliseconds.

Disc
Average disconnect time in milliseconds.

This data pertains to the current RMF interval. The IDEV command displays when the last RMF interval started, and how much of the current interval elapsed.

```
 DIDEV  Interval Start Time: 13:14:59     Elapsed:  4:50 MN
  PDSK  *        <=== Volser pattern
 +   Unit Volser  I/O per second  Util% Avg.Q   Resp = IOSQ + Pend + Conn + Disc
 +   ---- ------   Total    IMS   ----- ----- ------ ------  -----   ----   ----
 +    141 DISK01     7.8    1.2     3.1     3   14.0    3.0    2.0    8.0    1.0
 +    756 DISK02     5.1     .3      .3     1   22.4    1.0    2.9   15.5    3.0
```

Figure 86. PDSK display of devices allocated to the DLS address space

You can analyze devices that are allocated to various control region address spaces. In addition, you can analyze devices that are allocated to dependent regions. Use an RGNc command to select the target-dependent region. OMEGAMON for IMS analyzes devices that are allocated to the first region.

The statements in the following table show how to filter control regions.

Table 10. Control region address spaces

Statement	Target region
iIDEV *	control region
fIDEV *	control region
dIDEV *	DLS address space
sIDEV *	DLS address space
rIDEV *	DBRC address space
cIDEV *	DBRC address space
lIDEV *	IRLM address space
mIDEV*	IRLM address space
oIDEV *	volumes used for the current OLDS
wIDEV *	volumes used for the current WADS

The following statement shows how to analyze dependent region devices.

```
 RGND       IMSMPP01  IMSMPP03
PIDEV
 PDSK *     devices for region IMSMPP01
```

Figure 87. Dependent region devices

The following statement shows how to analyze DASD logging of OLDS devices.

```
OIDEV  Interval Start Time:  9:59:59     Elapsed:  2:14 MN
 PDSK  *        <=== Volser pattern
+   Unit Volser  I/O per second  Util% Avg.Q   Resp = IOSQ + Pend + Conn + Disc
+   ---- ------   Total    IMS   ----- ----- ------ ------  -----   ----   ----
+    156 DISK01     4.8     2.1   13.0   .02   27.8    3.8    4.5   12.8    6.7
+    153 DISK04     3.5     1.3    6.6   .01   25.0    3.0    5.1   15.2    1.7
```

Figure 88. Typical analysis of OLDS activity by using IDEV

The following statement shows how to analyze DASD logging of WADS devices.

```
WIDEV  Interval Start Time:  9:59:59     Elapsed:  2:14 MN
 PDSK  *        <=== Volser pattern
+   Unit Volser  I/O per second  Util% Avg.Q   Resp = IOSQ + Pend + Conn + Disc
+   ---- ------   Total    IMS   ----- ----- ------ ------  -----   ----   ----
+    159 DISK02    11.1     5.5    7.8   .01   24.3    3.8   20.5 0.00.0
```

Figure 89. Typical analysis of WADS activity by using IDEV

Chapter 7. Group definition and activation

Transaction groups or terminal groups can target the information that OMEGAMON for IMS collects and reports. After you set groups, you can target service level analysis for terminals and applications, and you can focus your tuning efforts on the specific bottlenecks that affect your system.

Defining groups

You can use the SETG command to change transaction and terminal group assignments dynamically.

To assign groups permanently, define the groups in the KIPGLB member in the RKIPGLBL DD data set. For information about the KIPGLB global parameters member, see the *IBM Tivoli OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

SETG

Provides a list of one or more criteria for inclusion of a transaction or node in a transaction group, a logical terminal group, or a node group.

Type:

Immediate

To determine whether a transaction belongs in a group, OMEGAMON for IMS compares certain characteristics of the transaction against characteristics that you specify for the group when you issue one or more SETG commands. The characteristics that SETG can match against are as follows:

- Transaction name or partial name
- PSB name or partial name
- Transaction scheduling class (not applicable to BMP)
- Logical terminal (LTERM) name or partial name
- VTAM node name

Groups are transaction-related, terminal-related, or node-related. Transaction-related groups contain transaction names, PSB names, or transaction classes. Terminal-related groups contain one or more logical terminal names. Node-related groups contain one or more VTAM node names; a node name is the logical unit name of a terminal as it is known to VTAM.

Issue the xSETGnn immediate command, which has the following format that is shown in Figure 90 on page 136.

SETG command format


```
    - or -
 xSETGnn  PSB=<PSB names....>               (names are 1-8 characters)
          - or -
 xSETGnn  CLASS=<transaction classes....>   (valid class range is 1 to 255;
                                             class range is defined in IMS gen)
          - or -
 xSETGnn  TERM=<logical terminal names....> (names are 1-8 characters)
          - or -
 xSETGnn  NODE=<VTAM node name....>         (names are 1-8 characters)
          - or -
 xSETGnn  no keyword, if only listing a transaction group's contents.
 |    |
 |    |
 |    |
 |    +------>  transaction group number (01-30 or max group # defined at
 |                                        installation time or via the MAXG
 |                                        command.  Use the number 99 to list,
 |                                        add, or delete from all groups.)
 |
 +----------->  A  -  add an entry to group nn
                C  -  create group nn
                D  -  delete an entry from group nn
                L  -  list group nn
                X  -  delete all entries from group nn
            blank  -  list group nn
```

Figure 90. SETG immediate command

x

Select one of the following options for the label:

- A - add an entry to group nn
- C - create group nn
- D - delete an entry from group nn
- L or blank - list group nn
- X - delete all entries from group nn

Specify a transaction group number from 01-30 or max group as defined at installation time or by using the **MAXG** command. Use the number 99 to list, add, or delete from all groups.

nn

Specify a transaction group number from 01-30, or max group as defined at installation time or by using the **MAXG** command. Use the number 99 to list, add to, or delete from all groups.

PSB

Indicates the PSB name, which can be 1 - 8 characters in length.

CLASS

Indicates the transaction class. A valid class range is predefined between 1 - 255.

TERM

Indicates the logical terminal name, which can be 1 - 8 characters in length.

NODE

Indicates the VTAM node name, which can be 1 - 8 characters in length.

No keyword

Leave blank, that is, specify no keyword, only if you are listing the contents of a transaction group.

When SETG completes a request to add, delete, or change, the command converts the label field to L (list) so that SETG does not re-execute the change on the next cycle.

You can specify multiple criteria for a group. You can, for example, specify a list of transaction names and a list of scheduling classes. You cannot, however, specify multiple criteria within the same command; you must issue separate commands for each criterion. In such cases, the group includes a transaction if it matches any one of the criteria. (The relationship between the criteria is OR, not AND.)

The SETG command allows generic transactions, PSB names, and logical terminal names. Generic node names are not allowed. When an asterisk (*) displays in an ID name, SETG accepts any character in that position. For example, the command

```
CSETG01 TRAN=*AR*
```

causes transaction group 01 to consist of all transactions whose name contains the letters AR in position two and three. Any character can fill positions one and four.

A trailing asterisk matches all remaining characters. Therefore, entering TRAN=*AR* is the same as entering TRAN=*AR*****.

Given the previous example, the totals for transaction group 01 include both transactions PART and MARK01.

You cannot modify any transaction groups via the SETG command (such as, adding or deleting transaction names) while RTA is active. You must first turn off RTA with the IRTA OFF command. You can, however, modify groups while DEXAN is active.

Any transaction or terminal group definitions and alterations that you make by using the SETG command only last during the current OMEGAMON for IMS session. You can set up screen spaces that contain any group definitions that you want to use again. For information about how to define or modify these groups permanently, see the description of the KIPGLB member in "Activating new groups" on page 140.

An individual group can be transaction-related, terminal-related, or node-related. If you try to add an incompatible item, OMEGAMON for IMS displays an error message and ignores your update request.

For transaction-related groups, a single SETG command accepts only one of the following keywords: TRAN=, PSB=, or CLASS=. A combination of these commands (shown as follows) does not produce the required result:

```
CSETG01  TRAN=PART, PSB=DFSSAM05
```

The appropriate command input is as follows.

```
CSETG01  TRAN=PART
ASETG01  PSB=DFSSAM05
```

SETG works the same way for terminal groups; use the TERM= or NODE= parameter instead of TRAN=, PSB=, or CLASS=.

Creating a group

You can use the CSETG command to create a group or a transaction group.

Procedure

- Issue the SETG command with the letter *C* in the label field.
- To create a transaction group that contains transaction names PART, ADDPART, and DLT0, enter:

```
CSETG03 TRAN=PART,ADDPART,DLT0
```

OMEGAMON for IMS alphabetizes the names and responds as follows:

```
LSETG03 TRAN=PART,ADDPART,DLT0
 +   Tran=ADDPART DLT0 PART
```

Figure 91. Transaction names

Note: If RTA or DEXAN is active, OMEGAMON for IMS rejects the CSETG command. OMEGAMON for IMS clears the bottleneck analysis buckets but does not create the group.

Adding entries to a group

You can use the ASETG command to add transactions or terminals to an existing group.

Procedure

- To add transactions or terminals to an existing group, issue the SETG command with the letter *A* in the label field.
- To add transactions to transaction group 03, which already contains the transaction names ADDPART, DLT0, and PART, issue the following command:

  ```
         ASETG03 TRAN=DLETPART
  ```

  OMEGAMON for IMS acknowledges your request and displays the IDs of all transactions in the redefined transaction group, as shown in the following figure.

  ```
  LSETG03 TRAN=DLETPART
  +   Tran=ADDPART DLETPART DLT0 PART
  ```

  Figure 92. Transaction IDs
- When you add new transaction names, PSB names, or transaction classes to a transaction group, a transaction group can consist of any combination of transaction names, PSB names, and transaction classes. If, for example, you want to add all class 05 transactions to the transaction group 03 defined previously, you can enter the following command:

  ```
          ASETG03 CLASS=5
  ```

  OMEGAMON for IMS acknowledges your request as shown in the following figure.

  ```
  LSETG03 CLASS=5
  +   Tran=ADDPART DLETPART DLT0 PART
  +  Class=5
  ```

  Figure 93. ASETG03 acknowledgement
- To make the add request effective for all transaction groups, specify a group number of 99 in the SETG command argument.

  For example, to add a transaction name to all transaction groups, use the following SETG command format.

  ```
          ASETG99 TRAN=ADDPART
  ```

Deleting entries from a group

You can use the DSETG command to delete transactions or terminals from a group.

Procedure

- To delete transactions or terminals from a group, issue the SETG command with the letter D in the label field.
- To delete transactions from a group, issue the command as shown in the following example.

  ```
         DSETG03 TRAN=ADDPART,DLT8
  ```

  OMEGAMON for IMS acknowledges your request and displays the IDs of all transactions in the redefined transaction group, as follows.

```
DSETG03 TRAN=ADDPART,DLT8
+ ID(s) not found - DLT8
+   Tran=DLETPART DLT0 PART
+  Class=5
```

Figure 94. DSETG03 acknowledgement

If you attempt to delete a transaction from a transaction group that does not include that transaction, OMEGAMON for IMS displays an error message that is similar to the “not found” statement that is shown in the previous figure.

- To delete a transaction from all transaction groups in which it displays, specify a group number of 99 using the following SETG command format.

```
DSETG99  TRAN=DLT0
```

- To delete all the transaction names, PSB names, and transaction classes from a transaction group, issue the SETG command with the letter X in the label field.

Listing the entries in a group

Use the LSETG command to list the transactions or terminals in a group. If DEXAN or RTA is active, the LIST function is the only function available with the SETG command

Procedure

- To list the transactions or terminals in a group, issue the SETG command with the letter L in the label field.
- To list the transactions or terminals in group 03, issue the command as follows:

```
LSETG03
```

OMEGAMON for IMS acknowledges your request and displays the names of all the transactions in the transaction group that you select, as shown in the following figure.

```
LSETG03
+   Tran=DLETPART PART
+  Class=5
```

Figure 95. LSETG03 acknowledgement

Note: To obtain the same results, you can also enter a blank space in the label field of the previous command instead of the letter L.

- To display the current contents of all the transaction groups, issue the following command:

```
LSETG99
```

- To request a list of all the groups that have defined entries, use a group number of 99.

The following figure shows the typical display output from the LSETG99command.

```
LSETG99
+ Group #1      (Name=GROUP 01)
+  Class=1
+ Group #3      (Name=BILLING )
+   Tran=DLETPART PART
+  Class=5
```

Figure 96. LSETG99 acknowledgement

The previous display indicates that only transaction groups 1 and 3 have defined entries. The display does not include groups, such as transaction group 2, above which no entries are defined (either transaction names, PSB names, transaction classes, or terminal or node names).

You can obtain the same result by entering a blank space in column 1 of the previous command instead of the letter L.

Specifying the maximum number of transaction groups

Use the MAXG command to specify the maximum number of transaction groups.

MAXG
Dynamically controls the number of transaction groups that OMEGAMON for IMS supports.

Type:
Immediate

Format:
MAXG nn

The value *nn* is the number of transaction groups that OMEGAMON for IMS is to support. The valid range is 1 - 30 groups. You can also define the maximum number of transaction groups the KIPGLB global parameters member is to support.

If you enter the MAXG command without any operands, the current maximum group value displays in a display similar to the following.

```
MAXG  >> 15 is the maximum number of transaction groups allowed <<
```

You cannot define a MAXG value lower than a currently active group ID number. If there are no active group ID numbers higher than the new MAXG value, the new value replaces the current maximum group number. If there are group IDs, which are larger than the new value, the maximum group number does not change, and a warning message displays as shown in the following example.

```
MAXG  5
+ max groups value not changed; following groups exceed MAXG value specified
+ groups= 6  7 10 14
+ enter an X in column 1 to delete these groups and change MAXG value
+      >> 15 is the maximum number of transaction groups allowed <<
```

Figure 97. MAXG warning message

This display shows that the current maximum group value is 15 and that transaction groups 6, 7, 10, and 14 contain entries. If you still want to reduce the maximum transaction group number to 5, place an X in the label field of the MAXG command as follows.

```
       XMAXG  5
```

The MAXG command then processes, and displays similar results to the following output:

```
XMAXG  5
+ following transaction groups have been deleted
+ groups= 6  7 10 14
+      >> 5 is the maximum number of transaction groups allowed <<
```

Figure 98. XMAXG results

Activating new groups

When you have multiple KIPGLB global parameters members in your RKIPGLBL DD data set, you can activate the default definitions in another KIPGLB member. The GLBL command provides this capability. In addition to reloading a new version of the current KIPGLB member (same suffix), you can also dynamically switch between multiple KIPGLB members, each with a different suffix.

When OMEGAMON for IMS starts, it loads default global definitions from the KIPGLBxx member in the RKIPGLBL DD data set, where *xx* is the value of the GLOBAL parameter in the startup procedure. KIPGLB is a source member that contains the default definitions for the transaction groups. It is created when you run PARMGEN to configure a runtime environment, but you can also create additional KIPGLB members. For more information about KIPGLB, see the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide*.

Rather than changing the current definitions dynamically by issuing the SETG and MAXG commands, you can use the GLBL to load another KIPGLB member that has the definitions that you want to use.

GLBL

Displays or sets the two-character suffix for the KIPGLB member name.

Type:

Immediate

Format:

GLBL*xx*

xx

Specifies the suffix.

If you enter the GLBL command without any operands, the suffix of the KIPGLB member currently in use displays. The following is a sample GLBL command output display:

```
 GLBL   >> 99  is the current KIPGLBxx suffix <<
```

From this display, you can see that the global parameters member currently in use is KIPGLB99. If you want to reload a new copy of this member or load a different member, enter the following command:

```
 GLBLxx
```

where *xx* is the suffix of the KIPGLB member that you want to load.

OMEGAMON for IMS responds to the previous command with the following message:

```
>GLBLMJ >> xx  is the current KIPGLBxx suffix <<
```

Note: When a > displays in column 1, the command comments out, preventing re-execution of the command.

If OMEGAMON for IMS is unable to locate the KIPGLB member with the specified suffix in the data sets allocated to the RKIPGLBL DD statement, the GLBL member does not change, and the following error message displays.

```
>GLBLMM >> OI901: Unable to read global KIPGLBMM check joblog for messages
```

Selecting groups automatically

The AUTO command is an immediate command that automatically defines monitoring groups. The AUTO command scans IMS control blocks and identifies the transactions or terminals that you use most frequently.

OMEGAMON for IMS automatically assigns these transactions or terminals to the first *n* monitoring groups, one transaction, or terminal per group. Issue the AUTO immediate command, which has the following format that is shown in Figure 99 on page 142.

AUTO command format

```
AUTO {x or t} o,n,FORCE
       ------  - - -----
         |     | |   |
         |     | |   +->Force option (Valid only with SET)
         |     | |
         |     | +----->transaction group range numeric option (1-MAXGRP)
         |     |         (Valid only with SET)
         |     |
         |     +-->Option = Set        - Establish automatic group selections
         |                = Reset      - Rescan for most active transactions
         |                               or terminals
         |                = Clear      - Clear groups 1-n
         |                = List       - List groups 1-n
         |                = ? or blank - List groups 1-n
         |                = Off        - Disable automatic monitoring group
         |                               selections
         |                = Test       - Identify 30 most frequent transactions
         |                               or terminals (Do not set groups)
         |
         +--->Type:  x = transactions (default)
                     t = terminals
```

Figure 99. AUTO command

x or t
Specifies the following types:

- x indicates transactions, which are the default value
- t indicates terminals

o
Specifies the following options:

- Set - establish automatic group selections
- Reset - scan again for most active transactions or terminals
- Clear - clear groups 1-n
- List - list groups 1-n
- ? or blank - list groups 1-n
- Off - Disable automatic monitoring group selections
- Test - Identify 30 most frequent transactions or terminals. Do not set groups

n
Indicates the transaction group range numeric option, where n is 1-MAXGRP. This parameter is valid only with SET.

FORCE
The FORCE parameter is valid only with SET.

Note: You cannot use this command when the RTA collector is active.

Creating groups automatically

You can use the AUTO SET command to automatically assign *n* transactions to a range of groups. The AUTO SET command specifies the AUTO range.

The AUTO command selects transactions or terminals by total use count (since startup) at the time you issue the command. OMEGAMON for IMS selects the *n* transactions or terminals that you use most frequently and assigns them to *n* groups in descending order (OMEGAMON for IMS assigns the transaction that you use most frequently to group 1).

The selection process proceeds only if the groups within the group range are empty. This is necessary to protect you from accidentally reassigning the monitoring groups. To override the test for empty groups, specify FORCE on the AUTO SET command.

```
       AUTO SET,5
```

OMEGAMON for IMS responds as shown in the following example:

```
LAUTO SET,5
+   Tran=CNX00220
+   Tran=CNX00230
+   Tran=CNX00240
+   Tran=CNX00250
+   Tran=CNX00260
```

Listing the entries in a group automatically

You can use the LAUTO command to display the contents of the groups that the automatic selection group range defines. The AUTO ? command is an alias for the AUTO list command

The command format is shown as follows:

```
LAUTO ?
Group 1:
+   Tran=CNX00260
Group 2:
+   Tran=CNX00250
Group 3:
+   Tran=CNX00240
Group 4:
+   Tran=CNX00230
Group 5:
+   Tran=CNX00220
```

Figure 100. AUTO list command

Resetting groups automatically

You can use the AUTO RESET command to reassign the most active transactions. You must start the SET function before you issue the AUTO command. Otherwise, an error message displays.

The command format is as follows:

```
       AUTO RESET
```

OMEGAMON for IMS responds with this output:

```
LAUTO RESET
+   Tran=CNX00200
+   Tran=CNX00210
+   Tran=CNX00220
+   Tran=CNX00230
+   Tran=CNX00240
+   Tran=CNX00250
+   Tran=CNX00260
```

Figure 101. AUTO command

Clearing a group automatically

You can use the AUTO CLEAR command to clear the transaction groups in the automatic selection process. You must start the SET function before you issue the AUTO command. Otherwise, an error message displays.

The command format is as follows:

```
        AUTO CLEAR
```

OMEGAMON for IMS responds with this output:

```
DAUTO CLEAR
```

Freezing a group automatically

After you set and reset the automatic selection groups, you might want to freeze the groups, that is, turn off the LIST, RESET or CLEAR functions. Group assignments remain in effect. To freeze the groups, use the AUTO OFF command. You must start the SET function before you issue the AUTO command. Otherwise, an error message displays.

The command format is as follows

```
        AUTO OFF
```

OMEGAMON for IMS responds with this output:

```
OAUTO OFF
+      Automatic group facility is now off; use SET to restart
```

Chapter 8. IMS regions

You can use OMEGAMON for IMS commands to display information about IMS regions.

The architecture of IMS requires that multiple z/OS address spaces be assigned to each IMS system that you start. There is always a control region, and some combination of message processing regions (MPPs or MPRs), batch message regions (BMPs), and fast path regions (IFPs). The following topics refer to all address spaces as regions.

- "IMS region immediate commands" on page 145
- "IMS region major commands" on page 145
- "IMS region minor commands" on page 149

IMS region immediate commands

Use the ISAP immediate command to display SAP support and status information.

ISAP

Starts and stops SAP support from the command interface and displays the status of SAP support (active or inactive).

Type:

Immediate

Format:

ISAP ON|OFF|?

Note: **?** is the default. If you do not specify a parameter, the system assumes that you want to know the SAP status.

Results:

The following example shows the results of an ISAP command.

```
________________ KOIISAP  VTM    OI-II      /C  I51A  08/29/08  10:56:04   B

>ISAP ON
+     SAP Support started.
>ISAP ?
+     SAP Support is active.
```

Figure 102. ISAP command

In this example,

- The **ISAP ON** starts SAP support.
- The **ISAP ?** displays the status of SAP support.

IMS region major commands

An OMEGAMON for IMS session associates only with a single IMS system even if there is more than one IMS system and, therefore, more than one IMS control region, active at the same time. You can issue different major commands to select IMS regions, but only regions that belong to that system are selected.

For example, if you have two IMS systems that are called PROD and TEST, with OMEGAMON for IMS running in each system, the OMEGAMON for IMS terminal monitoring the PROD control region completely ignores any IMS region that is associated with the TEST system.

OMEGAMON for IMS recognizes only an address space as an IMS dependent region after it makes itself known to the control region by using the IMS SVC and has an assigned PST. There is a brief period during both dependent region initialization and termination where the address space exists but OMEGAMON for

IMS does not see it. After the IMS control region recognizes the dependent region, OMEGAMON for IMS can display it.

If a region major command selects the control region, it always displays first in the display. Dependent regions always display in the same order you assign them to PSTs, except for the RGNL major command. For RGNL, you supply a list of region names, and OMEGAMON for IMS preserves your order.

Note: The region major commands have a special relationship with the IDEV (device activity analysis) command. To analyze the devices that are allocated to the first region in the list, which the region major command displays, use the P label on IDEV minor commands. Figure 103 on page 146 shows an example of where the RGN command is used with the IDEV command, and the activity of all devices that is allocated to region IMSMPP01 is displayed.

```
RGNL   IMSMPP01

  IDEV  Interval Start Time: 13:29:01     Elapsed: 12:27 MN    Length: 14:58 MN
 Ppdsk  *         <-- Volser Pattern
 +        Volser  I/O per second  Util% Avg.Q   Resp = IOSQ + Pend + Conn + Disc
 +        ------   Total     IMS  ----- ----- ------ ------ ------ ------ ------
 +    14B IMS100                     .1         20.1            .1    7.7   12.1
 +    161 IMS210
```

Figure 103. RGN command used with IDEV

BMP
Selects all batch message processing regions.

Type:
Major

BMP is an alias for RGNB.

CTL
Selects the IMS control region address space.

Type:
Major

CTL is an alias for RGNC.

DBRC
Selects the DBRC region only.

Type:
Major

DBRC is an alias for RGNR.

DLS
Selects the DL/I subordinate address space if LSO=S is in use.

Type:
Major

DLS is an alias for RGNS.

FSTP
Selects all fast path regions.

Type:
Major

FSTP is an alias for RGNF.

Note: This command is not applicable for a DBCTL system.

IRLM
Selects the IRLM region if the IRLM is in use instead of program isolation as the single lock manager.

Type:
Major

IRLM is an alias for RGNI.

MPP
Selects all message processing regions.

Type:
Major

MPP is an alias for RGNM.

Note: This command is not applicable for a DBCTL system.

PNR*nnn*
Provides access to all the *RGNx* minor commands for every DBCTL thread for thread analysis.

Type:
Major

The *nnn* parameter is the RGID of a thread that is displayed with the THRD command.

PNR displays the jobname of the CCTL associated with the thread identified by the PST. However, output is generated by entering the **`.EXM`** immediate command or any of the RGN*x* minor commands with PNR.

RGNA
Selects all IMS regions, including DBRC, DLISAS, and IRLM.

Type:
Major

Dependent regions include message processing regions (MPRs), batch message processing regions (BMPs), and fast path regions (IFPs).

RGNB
Selects all batch message processing (BMP) regions.

Type:
Major

BMP is an alias for RGNB.

RGNC
Selects the IMS control region address space.

Type:
Major

CTL is an alias for RGNC.

RGND
Selects all IMS dependent regions.

Type:
Major

OMEGAMON for IMS displays all IMS regions except the control region.

RGNE
Selects all non-idle IMS dependent regions.

Type:
Major

OMEGAMON for IMS displays all non-idle dependent regions, including DBCTL threads.

RGNF
Selects all fast path regions (IFPs).

Type:
Major

FSTP is an alias for RGNF.

Note: This command is not applicable for a DBCTL system.

RGNH
Selects all IMS BMP regions by using High Speed Sequential Processing (HSSP).

Type:
Major

RGNI
Selects the IRLM region if IRLM is in use instead of PI as the single lock manager.

Type:
Major

IRLM is an alias for RGNI.

RGNL
Specifies up to seven IMS regions by job names.

Type:
Major

OMEGAMON for IMS automatically eliminates any names in the list, which are not valid. OMEGAMON for IMS considers address spaces that are associated only with the IMS control region as valid. Regions display in the same order in which you list them.

RGNM
Selects all message processing regions (MPPs).

Type:
Major

MPP is an alias for RGNM.

Note: This command is not applicable for a DBCTL system.

RGNN
Selects all non-dependent IMS regions.

Type:
Major

Non-dependent regions include the IMS control region, the DBRC region, the DLISAS or DLS region, and the IRLM region.

RGNP/n
Selects all regions that match a pattern.

Type:
Major

The **..SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If RGNP does not find a /n value, it uses the first pattern that is supplied with the last **.SPT** command.

RGNR
Selects the database recovery control region (DBRC).

Type:
Major

DBRC is an alias for RGNR.

RGNS
Selects the DL/I subordinate address space if LSO=S is in use.

Type:
Major

DLS is an alias for RGNS.

RGNT
Selects the RSR Transport Manager Subsystem address space.

Type:
Major

TMS is an alias for RGNT.

RGN2
Selects all IMS regions that have external subsystems, which are defined.

Type:
Major

IMS region minor commands

For any region that you select by issuing a major command, issue the associated minor commands to obtain more details. To list minor commands for a major command, enter the `.MIN` minor command after the major command.

The following three categories of minor commands apply.

- Commands that relate to the IMS environment.
- Commands that provide information that is related to the z/OS environment in which the region runs.
- Commands that relate to fast path regions.

IMS-related region minor commands

You can use many minor commands to display information about IMS-related regions. Some of these minor commands, however, do not apply to all types of IMS regions.

For example, IMS cannot schedule a PSB into the control region. The message --n/a-- indicates that the minor command does not apply to the region selected by the major command.

AUTH
Displays the total number of DL/I message AUTH calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

CALL
Displays user parameter information for the first region displayed.

Type:
Minor of IMS region majors

Format:
cCALL

H in column 1 displays the I/O PCB in hexadecimal and character format.

CALL displays the call function code, the first 60 bytes of the current PCB, and up to 60 bytes of the I/O area. DB calls also show segment search arguments.

This command does not apply to the control region, non-dependent regions, or fast path regions.

The following figure shows the output of a typical CALL command.

```
CALL       User Parm Information For Current DLI Call
+      FUNC  GU
+      PCB   DCLOOP  ..
+      IOA   ....DCLOOP
+      SSA1  BE3PARTS (GEORGE)
+      SSA2  BE3PARTS (SAM)
```

Figure 104. CALL command display

CDMB
Displays the name of the most recently referenced database (DMB).

Type:
Minor of IMS region majors

This command does not apply to the control region.

CHNG
Displays the total number of DL/I message CHNG calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

CLAS
Displays the IMS scheduling class for a transaction that is currently scheduled in a region.

Type:
Minor of IMS region majors

This command does not apply to the control region.

CMD
Displays the total number of DL/I message CMD calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

CPCB
Displays the TCB processor usage percentage that is normalized over the number of online processors.

Type:
Minor of IMS region majors

This command does not apply to the control region.

CTRM
Displays the current CICS® terminal id for this DBCTL thread. This information is retrieved from the CICS EXEC Interface Block in the CICS address space that runs the DBCTL thread.

Type:
Minor of RGN majors and their aliases.

CTRN
Displays the CICS transaction name for the current thread.

Type:
Minor of RGN majors and their aliases.

CTSK
Displays the current CICS task number for this DBCTL thread. This information is retrieved from the CICS EXEC Interface Block in the CICS address space that runs the DBCTL thread. The task number can be used in a CICS operator command to cancel the unit of work for such reasons as a looping PSB or a deadly embrace situation.

Type:
Minor of RGN majors and their aliases.

DBT
Displays the total number of DL/I database calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region or fast path regions.

DEQ
Displays the total number of DL/I database DEQ calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

DGN
Displays the total number of DL/I database GN calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region.

DGU
Displays the total number of DL/I database GU calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region.

DLET
Displays the total number of DL/I database DLET calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region or fast path regions.

DPCB
Displays the name of the most recently referenced database PCB.

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or fast path regions.

DRCS
Displays the size of the dependent inter-region communication area (DIRCA).

Type:
Minor of IMS region majors

The size is in bytes (decimal). This command does not apply to control region or other non-dependent regions.

ESAF
Displays the total number of external subsystem calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

ETIC
Displays the elapsed time for waiting on intent conflicts.

Type:
Minor of RGN majors and their aliases.

ETIM
Displays the elapsed time of the transaction that is currently running, in microseconds.

Type:
Minor of IMS region major

ETIO
Displays the elapsed time for waiting on database I/O.

Type:
Minor of RGN majors and their aliases.

ETLK
Displays the elapsed time for waiting on database locking.

Type:
Minor of RGN majors and their aliases.

ETPL
Displays the elapsed time for waiting on pool space.

Type:
Minor of RGN majors and their aliases.

ETSP
Displays the elapsed time that is incurred for the schedule process.

Type:
Minor of RGN majors and their aliases.

FLD
Displays the total number of DL/I database FLD calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

GCMD
Displays the total number of DL/I message GCMD calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

GHN
Displays the total number of DL/I database GHN calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or fast path regions.

GHNP
Displays the total number of DL/I database GHNP calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or fast path regions.

GHU
Displays the total number of DL/I database GHU calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or fast path regions.

GMSG
Displays the total number of DL/I system service GMSG calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

GNP
Displays the total number of DL/I database GNP calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or fast path regions.

HSPB
Displays HSSP private area buffer pool byte statistics.

Type:
Minor of IMS region majors

Displays HSSP private area buffer pool statistics for the first region that is displayed by the major command. The display fields are as follows.

Area
Displays the DEDB area that HSSP is processing.

Alloc
Displays the total number of bytes that were allocated for the private area buffer pool.

Used
Displays the number of bytes used by the private area buffer pool.

%Used
Displays the percentage of bytes used in the private area buffer pool.

ICAL
Displays the total number of DL/I message ICAL calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

ICMD
Displays the total number of DL/I system service ICMD calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

IHLD
Displays the number of IRLM or PI locks held.

Type:
Minor of RGN majors and their aliases.

INIT
Displays the total number of DL/I system service INIT calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

INQY
Displays the total number of DL/I system service INQY calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

INUM
Displays the number of databases I/Os for the transaction instance (UOR).

Type:
Minor of RGN majors and their aliases.

IOPS
Displays the total number of databases I/Os per schedule.

Type:
Minor of RGN majors and their aliases.

ISRT
Displays the total number of DL/I database ISRT calls for the transaction instance (UOR).

Type:
Minor of RGN majors and their aliases.

This command does not apply to the control region, other non-dependent regions, or fast path regions.

LOCK
Displays PI or IRLM locks currently held by a dependent region.

Type:
Minor of RGN majors and their aliases.

The following figure shows an example of the LOCK command display.

```
 RGND  MPP00131 MPP00121 MPP00132 MPP00150
 lock  Subsys   Workunit PSBname  Tx/Rg ID  Lterm ID  Status DB/Area  Token    DCB
+      IMSA     MPP00121 ACCNT010 UPDACCT   R105A10   UP/OWN ACCNTDBA 00004192  1
+      IMSA     MPP00131 UPDCUST1 UPDCUST   L050C09   UP/OWN ACCNTDBA 00009566  1
```

Figure 105. LOCK command display

A description of the fields in the LOCK display follows.

Subsys
Displays the name of the subsystem that holds the lock.

Workunit
Displays the name of the job that holds the lock.

PSBname
Displays the program specification block (PSB) associated with the lock.

Tx/Rg ID
Displays the transaction name or region number (if DBCTL is monitored) associated with the lock.

Lterm ID
Displays the logical terminal name that is associated with the lock.

Status
Displays the status of the job (workunit) holding the lock and indicates whether it is an owner or a waiter.

These statuses are as follows:

EX
Exclusive control

UP
Update

RD
Read

SH
Share

ER
Erase

Owners and waiters

WAT
Workunit is waiting for access.

OWN
Workunit owns the lock.

DB/AREA
Database name, DEDB area name, or partition name.

Token
Displays the relative byte address/relative block number (PI only).

DCB
Displays the DCB number within the named DMB (PI only).

DBName
Database name or DEDB area name.

PartName
Partition name (if partitioned).

Part.ID
Partition ID (if partitioned).

n/a displays if data are not available because there is more than one IRLM and the secondary IRLM is running on a different processor.

Only those locks that are held by the first region that is displayed by the major command are shown. For example, in Figure 105 on page 154, the lock for region MPP00131 is the only one displayed. To display the locks for the other dependent regions, use RGNL.

LOG
Displays the total number of DL/I system service LOG calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

MGN
Displays the total number of DL/I message GN calls for the transaction instance (UOR).

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, fast path regions, or DBCTL threads.

MGU
Displays the number of DL/I message GU calls for a transaction instance (UOR) which is 1 for IFPs, MPPs, and message driven BMPs. This count might be the result of a CHKP call, which returns the next input message into the I/O area for IFPs, MPPs, and message driven BMPs.

MIRT
Displays the total number of DL/I message ISRT calls for the transaction instance (UOR).

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, fast path regions, or DBCTL threads.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, fast path regions, or DBCTL threads.

MNUM
Displays the I/O PCB message number that is assigned to the PST associated with the current region.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, fast path regions, or DBCTL threads.

MPRG
Displays the total number of DL/I message PURG calls for the transaction instance (UOR).

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, fast path regions, or DBCTL threads.

MSGT
Displays the total number of DL/I message calls for the transaction instance (UOR).

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or fast path regions.

OCUP
Displays the occupancy for a message region or DBCTL thread.

Type:
Minor of IMS region major

The occupancy displays as a percentage.

To calculate this number, OMEGAMON for IMS samples the message regions or DBCTL threads during every OMEGAMON for IMS cycle and observes the number of times it finds an active transaction or thread. (It does not consider a region in WFI wait active, which means that WFI regions display as less than 100% occupied.) For the first 120 samples (or about ten minutes when the interval is five seconds) the data is statistically insignificant, therefore OMEGAMON for IMS displays **--init--.**

After initialization is complete, the numbers represent message region occupancy or thread occupancy averaged over the last hour.

OCUP does not calculate region occupancy percentages for BMPs.

OIOR
Displays the total number of OSAM reads for the transaction instance (UOR).

Type:
Minor of IMS region majors

OTMI
Displays OTMA information for the first region displayed.

Type:
Minor of IMS region major

The OTMI minor command outputs only the OTMA information for the first (or only) region. OTMA information consists of the TMEMBER name, TPIPE name, commit mode, synchronization level, and TPIPE status.

PLAN
Displays the DB2® plan ID for the current transaction.

Type:
Minor of IMS region major

PLIM
Displays the current application processing limit.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

PLSC
Displays the number of transaction schedulings of the DB2 PLAN since the last IMS cold start.

Type:
Minor of IMS region major

If a transaction is not active, OMEGAMON displays **--none--**.

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

PLSY
Displays the number of DB2 commits during this schedule.

Type:
Minor of IMS region major

POS
Displays the total number of DL/I database POS calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

PSBN
Displays the PSB name of a transaction if one is scheduled.

Type:
Minor of IMS region major

If there is not a transaction in progress, OMEGAMON for IMS displays **--none--**.

This command does not apply to non-dependent regions.

PSTA
Displays the PST address.

Type:
Minor of IMS region major

QTME
Displays the elapsed time for the current transaction instance by using microsecond precision.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

RCMD
Displays the total number of DL/I system service RCMD calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

REPL
Displays the total number of DL/I database REPL calls for the transaction instance (UOR).

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or fast path regions.

RGID
Displays the IMS region identification number for the selected region.

Type:
Minor of IMS region major

This ID is the same as the PST number and the region number the IMS DC monitor refers to.

This command does not apply to non-dependent regions.

RLSE
Displays the total number of DL/I database RLSE calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

ROLB
Displays the total number of DL/I system service ROLB calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

ROLS
Displays the total number of DL/I system service ROLS calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

RTYP
Displays the type of the selected IMS region.

Type:
Minor of IMS region major

The region type can be BMP, Control, DLS, DBRC, Fast Path, HSSP-BMP, IRLM, or Message.

SALL
Displays the name of all defined external subsystems and their associated parameters for the first region displayed.

Type:
Minor of IMS region major

SALL displays the external subsystem name, current external subsystem interface status, the language interface token (LIT), resource translation table (RTT) name, interface control module name, and the error option specified.

The current external subsystem interface status values include Init, Iden, Echo, Trm-Iden, Term, Resolve, Sign on, Cre-Thrd, Trm-Thrd, Sign off, Prepare, Abort, Not-Oper, SQL-Call, MQ-Call, Command, and Commit. More information about these status values can be found in the PARM minor command of the external subsystem major commands.

SAPC
Displays the accumulated processor time in hundredths of a second for the SAP transaction currently running in a region. If the current transaction is not an SAP transaction the SAPC displays **Not SAP**.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

SCLS
Displays the *nn*th class of the selected IMS region.

Type:
Minor of IMS region major

You must specify the number *nn*, from one to four, in the argument field of the command. If the region does not have an *nn*th class that is defined, **--none--** is displayed. (**--n/a--** displays if the selected region is a non-dependent region, since scheduling classes are not applicable.)

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

SETC
Displays the total number of DL/I message SETO calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

SETO

Displays high speed sequential processing options for a region.

Type:

Minor of IMS region major

SETO displays information that is specified in the SETO control card in the DFSCTL data set. The display shows the following information:

dbname

Displays the database name that is specified in the SETO control card in the DFSCTL data set.

PCB

PCB name that is specified in the SETO control card in the DFSCTL data set.

#I/C

Displays the following image copy information:

1

One image copy

2

Dual image copy

0

No image copy

I/C-Opt

Displays the following image copy options:

CONTINUE

Specifies that the program continue if the image copy cannot complete.

1ABEND

Specifies that the program abend if the image copy cannot complete for one data set.

2ABEND

Specifies that the program abend if the image copy cannot complete for two data sets.

Area

Displays the optional parameter on the SETO control card that allows an image copy of an area of a DEDB.

SETR

Displays the DEDB areas for PCBs as specified in the DFSCTL SETR control statement.

Type:

Minor of IMS region major

SETR displays the processing areas of DEDBs to which an application program is restricted during scheduling. The application program can access only data in the DEDB within the area or areas specified.

The output fields in the SETR display are described in the following list.

dbname

Displays the database name.

PCB

Displays the PCB name.

Area

Displays the area.

SETS

Displays the total number of DL/I system service SETS calls for the transaction instance (UOR).

Type:

Minor of IMS region majors

SETU
Displays the total number of DL/I system service SETU calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

STAT
Displays the status of the IMS region.

Type:
Minor of IMS region major

The following list explains the possible status values.

`Region Inactive`

Idle
No application is scheduled in this region

Hot-RGN
Hot Region: idle, but primed with an application

`Scheduler Waits`

Wt-Intnt
Waiting because of intent conflict

Wt-DMB
Waiting for DMB to load

Wt-Mover
Waiting for block mover

Wt-Nowrk
Waiting for work (same as idle)

Wt-PSB
Waiting for PSB to load

Wt-WFI
WFI transaction is waiting for input

`IWAITs`

Wt-PIenq
IWAITing because of program isolation enqueue conflict

Wt-DL/I
IWAITing for DL/I processing

Wt-DISP
IWAITing in IMS dispatcher

Wt-LATCH
IWAITing for a latch

Wt-TERM
IWAITing in termination

IWAIT
Other IWAIT

`Execution States`

Ex-DL/I
Active in DL/I processing

Ex-Term
Active in termination

Ex-Drgn
Active in IMS dependent region

Wt-IRLM
Wait for IRLM lock conflict

Ex-Abend
Running in ABDUMP manager. In this execution state, canceling a dependent region might cause a U0113 abend.

Ex-LUM
Running in LU 6.2 manager. In this execution state, canceling a dependent region might cause a U0113 abend.

Ex-DB2
Active in the DB2 external subsystem

Ex-MQ
Active in the MQ external subsystem

Ex-ESS
Active in the external subsystem other than DB2 andMQ

`Fast Path Status`

Wt-FPWFI
Fast path transaction is waiting for input

Wt-FxBuf
Waiting for a fixed buffer

Wt-MSDB
Waiting for an MSDB segment

Wt-OCLth
Waiting for the OPEN/CLOSE LATCH

Wt-DmLth
Waiting for the DMAC LATCH

Wt-MSLth
Waiting for the MSDB LATCH

Wt-DEOwn
Waiting for ownership within a DEDB

Wt-OBA
Waiting for OBA interlock

Wt-SYLok
Waiting for the SYNC LOCK

Ex-Sync
Active in SYNC POINT processing

Ex-Drgn
Active in fast path dependent region

This command does not apply to non-dependent regions.

SYST
Displays the total number of DL/I system service calls for the transaction instance (UOR).

Type:
Minor of IMS region majors

TERM
Displays the logical terminal ID associated with a transaction that is scheduled into an MPR.

Type:
Minor of IMS region major

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

TIME
Displays the time the current transaction input is placed on the queue.

Type:
Minor of IMS region major

The I/O PCB indicates the time the TP transaction is placed on the queue. The time of transactions, which results from program insert, is the time of the originating TP transaction, not the time of the insert. OMEGAMON for IMS displays the time in the format *hh:mm:ss*.

This command does not apply if the region is waiting for input.

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

TOKN
Displays the 8-byte DBCTL thread recovery token in hexadecimal format.

Type:
Minor of RGN major

If the PST selected by the RGN major is not a DBCTL thread, **--n/a--** displays.

TPCB
Displays the name of the most recently referenced TP PCB.

Type:
Minor of IMS region majors

If the region is waiting on a message queue I/O, TPCB displays I/O PCB. If no TP PCB is referenced, OMEGAMON for IMS displays **none**.

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

TRAN
Displays the name of any IMS transaction that is scheduled into the IMS region.

Type:
Minor of IMS region majors

This command does not apply to the control region, other non-dependent regions, or DBCTL threads.

VIOR
Displays the total number of VSAM reads for the transaction instance (UOR).

Type:
Minor of IMS region majors

WTDE
For DBCTL threads only, displays the number of waits for DEDB buffers.

Type:
Minor of RGN majors and their aliases.

WTEE
Displays the number of waits on exclusive enqueues.

Type:
Minor of RGN majors and their aliases.

WTQE
Displays the number of waits on Q command code enqueues.

Type:
Minor of IMS region majors

WTTE
Displays the number of waits on test enqueues.

Type:
Minor of RGN majors and their aliases.

WTUE
Displays the number of waits on update enqueues.

Type:
Minor of RGN majors and their aliases.

XSID
Displays the external subsystem ID when an external subsystem request is in progress, as indicated by a status value (STAT minor) of Ex-DB2, Ex-MQ, or Ex-ESS.

Type:
Minor of RGN majors and their aliases.

XSST
Displays the external subsystem interface status when an external subsystem request is in progress, as indicated by a status value (STAT minor) of Ex-DB2, Ex-MQ, or Ex-ESS.

Type:
Minor of RGN majors and their aliases.

The external subsystem interface status values include Init, Iden, Echo, Trm-Iden, Term, Resolve, Sign on, Cre-Thrd, Trm-Thrd, Sign off, Prepare, Abort, Not-Oper, SQL-Call, MQ-Call, Command, and Commit. More information about these status values can be found in the **PARM** minor command of the external subsystem major commands.

z/OS-related region minor commands

You can use many minor commands to display information about z/OS-related regions.

ACTI
Displays the elapsed time since the region began. The elapsed time is the time in DDD:HH format since the address space is started. For long-running tasks, use the ACTI command, instead of the TMTR command.

Type:
Minor of IMS region majors

If the region start time is not available, this field displays `Unavail` and one of the following reason codes.

Note: Because of space constraints, the reason code replaces the `l` in `Unavail`, (for example, `Unavai7`).

- 1 = NO ASCB POINTER
- 2 = INVALID ASCB EYECATCHER
- 3 = NO ASCB JOBNAME
- 4 = ASCB JOBNAME DOES NOT MATCH PST ENTRY TABLE JOBNAME
- 5 = NO TCB POINTER
- 6 = NO JSCB POINTER
- 7 = NO JCT POINTER
- 8 = NO JMR POINTER
- 9 = INVALID JMR EYECATCHER
- 0 = REGION START TIME > 24 HOURS

AENV
Displays graphic environmental data about a region.

Type:
Minor of IMS region majors

AENV displays graphic environmental data about a region over both the last interval and the last 20 intervals.

When the major command selects more than one region, AENV operates only on the first region it selects; it ignores the rest.

The example that follows shows the output of a sample AENV command for the IMSPROD control region.

```
 RGNC   IMSPROD
 aenv  SYSTEM DATA- SHORT 0.1.2.3.4.5.6.7.8.9.0   LONG  0.1.2.3.4.5.6.7.8.9.0
+      TCB TIME(%)  25.3 |----->. . . . . . . .    5.5 |->. . . . . . . . . . .+
+      SRB TIME(%)   9.7 |->. . . . . . . . . .     .7 |> . . . . . . . . . . .+
+      PAGE-INS/S_   3.8 |------=>. . . . . . .     .1 |> . . . . . . . . . . .+
+      PAGE-OUTS/S   2.8 |----->. . . . . . . .     .0 |> . . . . . . . . . . .+
+      I/O'S/SEC__   7.2 |->. . . . . . . . . .    7.2 |->. . . . . . . . . .
+      SU'S/SEC___ 131.4 |--> . . . . . . . . .  105.4 |--> . . . . . . . . . .+
+      WORKING SET 1400K |--> . . . . . . . . .  1406K |--> . . . . . . . . . .-
+      PG-IN/CPU-S  12.5 |--> . . . . . . . . .   30.7 |------=>. . . . . . . .+
```

Figure 106. AENV command

The **SHORT** graph on the left side of the display represents the last OMEGAMON for IMS cycle; the **LONG** graph on the right side of the display represents a running average of the previous 20 OMEGAMON for IMS cycles. The last column of the display indicates:

```
        The trend is upward (+).
        The trend is downward (-).
        No change ( ).
```

ASID

Displays the z/OS address space ID of the region.

Type:
Minor of IMS region majors

The address space ID displays in hexadecimal.

z/OS numbers the regions that begin with the *MASTER* scheduler at ASID 0001. As z/OS starts regions, it assigns them an ASID number, which it uses as an internal reference.

CPU

Displays the TCB processor time for the current job step for the region in seconds.

Type:
Minor of IMS region majors

CPUP

Displays the TCB processor usage percentage that is normalized over the number of online CPUs.

Type:
Minor of IMS region majors

DISP

Displays the type of z/OS dispatching algorithm in use for a region.

Type:
Minor of IMS region majors

The algorithm type is one of the following types: MTW (mean-time-to-wait), ROTATE, TIME-SLC, or FIXED.

DOM#

Displays the current domain number of the region with which it is associated.

Type:
Minor of IMS region majors

The IEAIPSnn member of SYS1.PARMLIB defines and controls domains.

DPRT

Displays the processor dispatching priority of the region in both decimal and hexadecimal.

Type:
Minor of IMS region majors

FIXF
Displays the fixed frame count when a region is swapped out.

Type:
Minor of IMS region majors

Under ordinary circumstances, z/OS never swaps out IMS regions, but some users modify their IMS systems to allow it.

FMCT
Displays the number of frames a region is using in main storage.

Type:
Minor of IMS region majors

Note: For z/OS, expanded storage is included.

When a region runs under z/OS, portions of the program can be in main storage and other temporarily inactive portions can be on disk (on a PAGE or SWAP data set.) A unit of main storage (4096 bytes) is a frame. The unit of disk storage that holds one frame is a slot.

The frame count is zero when the region is swapped out. Instead of displaying zero, OMEGAMON for IMS displays the last frame count (and marks it with an S).

Regions can be resident and still show an S for FMCT, which usually means that the region is page-stolen down to zero.

FXFR
Displays the number of frames of real storage a region is using that are fixed and cannot move to disk (as slots) to make room for other regions.

Type:
Minor of IMS region majors

One frame equals 4096 bytes.

HUIC
Displays the highest unreferenced interval count for the region.

Type:
Minor of IMS region majors

A low number for HUIC indicates the system is paging heavily and there is a high demand for real storage frames.

IODP
Displays the I/O dispatching priority for the region.

Type:
Minor of IMS region majors

IOJ
Displays the I/O counts for the region.

Type:
Minor of IMS region majors

Use the IOJ command with the **.R** commandsuffix to show the I/O rate during the last OMEGAMON for IMS cycle.

JPCI
Displays the job common area page-in counts.

Type:
Minor of IMS region majors

Use with the **.R** command to display the common area page-in rate.

JPUI

Displays the job private area page-in counts.

Type:

Minor of IMS region majors

Use with the **.R** command to display the private area page-in rate.

JPUO

Displays the job private area page-out counts.

Type:

Minor of IMS region majors

Use with the **.R** command to display the private area page-out rate.

JSTA

Displays the job status indicator for the region.

Type:

Minor of IMS region majors

The following figure illustrates the JSTA command display.

```
RGNC  IMS210FP
jsta   I/O*NSW
```

The job status indicator (I/O*NSW, previous) contains three fields in the following format.

```
        aaabccc
```

aaa

Dispatchability:

CPU

Region is processor dispatchable.

WAT

Region is waiting.

DLY

Region is delayed.

I/O

Region completed I/Os but is not processor dispatchable.

b

Transaction flag:

*****

Region is in an z/OS transaction.

( )

Region is not in an z/OS transaction.

ccc

Location:

RES

Region is resident.

NSW

Region is resident and nonswappable.

LSW

Region is logically swapped.

SWP

Region is swapped out.

NVSC
Displays the number of non-VIO slots the region uses.

Type:
Minor of IMS region majors

These slots on a PAGE data set are used for portions of the program.

The example that follows shows the number of non-VIO slots each dependent region uses.

```
RGND  MESSAGE1 MESSAGE2    BMP03
fmct        40 S    30        10  frame count
wkst      160K S  120K       40K  working set size
fxfr        10                 5  fixed frames
nvsc        40      33        14  non-VIO (program)slots
vsc          5                    VIO slots
```

Figure 107. non-VIO slots

PERF
Displays the performance group number of a region.

Type:
Minor of IMS region majors

The PERF and DOM# minor commands display the performance group number and domain number of a region. The SRM manages performance in the z/OS operating system by swapping regions out and in according to defined rules. The SRM uses service units to track the amount of computer resources a region is using. These service units are a composite of processor time used, I/Os completed, and main storage occupancy for the region. After the activity in a region is examined, different parts of the SRM indicate whether IMSswaps the region in or out. (Usually, IMS marks all regions non-swappable unless the user circumvents it.)

The IEAIPScc member of SYS1.PARMLIB defines and controls performance groups.

PROC
Displays the current procedure step name for the region.

Type:
Minor of IMS region majors

PSTI
Displays a summary of parameters and data about storage isolation.

Type:
Minor of IMS region majors

The example that follows shows the output of a typical PSTI command.

```
 RGNL  MESSAGE1
 psti  Private:     Working         Page-ins
+                   Set Size        /CPU-sec
+      Maximum        100K           -none-
+      Target          60K           ------
+      Actual          72K             0.13
+      Minimum         40K           -none-
```

Figure 108. PSTI command

In this example, the MESSAGE1 region is in a performance group, which has a minimum of 40K and a maximum of 100K established as storage isolation limits. MESSAGE1's actual current working set size is 72K, but the current target size storage isolation set is 60K.

The actual value for page-ins/CPU-second is 0.13; '**-none-**' indicates that the value does not exist; '**------**' indicates that the field does not apply.

Note: OMEGAMON for IMS labels the /CPU-sec field as /Elap-sec when it runs in internal monitoring.

Only one PSTI display can be active on an OMEGAMON for IMS display at a time. If you attempt to display data from two different jobs at the same time, PSTI does not complete initialization.

SEQN

Displays the sequence number of the region on the processor dispatching queue.

Type:
Minor of IMS region majors

The next example shows the processor dispatching queue sequence number of each dependent region.

```
RGND   MESSAGE1 MESSAGE2     BMP01
cpu      144.23    58.09     78.32 <step CPU time in seconds>
cpup       .038     .117      .012 <percent TCB time used in last interval>
srbt      11.23    43.18     58.09 <SRB time for step in seconds>
srbp       .009     .023      .008 <percent SRB time used by region in interval>
tcpu     155.46   101.27    136.41 <SRB+TCB time in seconds>
tcpp       .047      .14       .02 <percent TCB+SRB time in last interval>
disp        MTW      MTW       MTW <dispatching algorithm>
dprt    (FA)250  (FC)252   (FC)252 <dispatching priority>
seqn         18       16        14 <sequence number on CPU dispatching queue>
```

Figure 109. CPU dispatching queue

SRBP

Displays the SRB processor usage percentage that is normalized over the number of online processors.

Type:
Minor of IMS region major

SRBT

Displays the SRB CPU time of the current job step for the region in seconds.

Type:
Minor of IMS region majors

STEP

Displays the current step name for the region.

Type:
Minor of IMS region majors

The example that follows shows the output of a typical STEP command.

```
RGND  MESSAGE1 MESSAGE2    BMP01
asid  (0C)  12 (15)  21 (18)  24
dom#         1        1       13
perf         6        6       18
proc                          GO
step    REGION   REGION ORDERUPD
```

Figure 110. Output of a typical STEP command

SUAL

Displays all service units for the period.

Type:
Minor of IMS region majors

All service units = SUCP + SUIO + SUMS.

SUCP

Displays the processor service units for this period.

Type:
Minor of IMS region majors

Use with the **.R** command to display service units per second during the last interval.

SUIO
Displays the I/O service units for this period.

Type:
Minor of IMS region majors

Use with the **.R** command to Display service units per second during the last interval.

SUMS
Displays the main storage occupancy service units for this period.

Type:
Minor of IMS region majors

Use with the **.R** command to display service units per second during the last interval.

TCPP
Displays the total (TCB + SRB) processor usage percentage that is normalized over the number of online CPUs.

Type:
Minor of IMS region majors

You can use the previous commands (CPUP, SRBP, and TCPP) to display the percent of the processor that this region is using. Percentages are displayed as decimal values; for instance, 2.7% is displayed as 0.027. This percentage value is displayed as taking into account all available processors, which means that the value never exceeds 100 percent.

TCPU
Displays the total CPU (SRB + TCB) time for the region in seconds.

Type:
Minor of IMS region majors

This must equal the sum of the **CPU** and **SRBT** commands.

TMTR
Displays the time since the region began.

Type:
Minor of IMS region majors

Transaction refers to an z/OS rather than IMS transaction. (An z/OS transaction begins every time the performance group changes.) This field displays the time since the last transaction began. If the performance group changes during execution of the job, the TMTR reflects the time since that change. For instance, it might measure only the time since a STEP change. Otherwise, it measures the total job time.

If you run the z/OS operator command SET IPS, IMS does not reset the times of transactions that are swapped out at the time of execution until they are swapped in again.

TWSS
Displays the target working set size in K (1024 bytes) for any fenced region.

Type:
Minor of IMS region majors

This command is for installations that use the z/OS feature called storage isolation.

If the maximum target working set size is reached, MAXIMUM displays.

VSC
Displays the number of VIO slots the region used.

Type:
Minor of IMS region majors

Virtual I/O (VIO) is a method of using virtual memory for temporary files.

WAIT
Displays the time since the region was last active.

Type:
Minor of IMS region majors

Active means the time since any processor was last expended.

WKST
Displays the working set size in number of K (1024 bytes) a region is using in main storage.

Type:
Minor of IMS region majors

WKST = 4 x FMCT.

The value is zero when the region is swapped out. Instead of displaying the zero value, OMEGAMON for IMS displays the last working set size (and marks it with an S). Regions can be resident and still show an S for WKST, which usually means that the region is page-stolen down to zero.

Examples

Figure 111 on page 170 and Figure 112 on page 170 show several typical **Region** major and minor commands.

```
RGND  MESSAGE1    BMP02    BMP03
jsta   CPU*RES  I/O*RES  WAT*SWP
ioj .R    24.1     12.1        0  <I/O rate>
iodp   (04)212   (63)99   (63)99  <I/O dispatching priority>
tmtr   3:20 MN  2:10 MN  1:15 MN  <time in transaction>
wait              1 SEC  1:23 MN  <time since CPU used>
```

Figure 111. Region major command with several minor commands

```
RGND    MPP01     BMP01     MPP05
sucp       50       182       883 <service units - CPU>
suio      130         0         0 <  "      "    - I/O>
sums                 27           <  "      "    - Main Storage>
sual      180       209       883 <  "      "    - Total>
huic        4        12        17 <highest unreferenced interval count>
jpui.R     .2        .7       5.1 <page-in rate - pages/second>
jpuo.R    1.0       2.2       2.1 <page-out rate - pages/second>
jpci.R              1.5           <common area page-in rate - pages/second>
```

Figure 112. Region major command with service unit and page rate minor commands

Fast path region minor commands

You can use the fast path region minor commands with the RGNF and FSTP major commands.

BFWT
Displays the number of times the region is idle because the buffer waits for free buffers for accessing a fast path database.

Type:
Minor of IMS region majors

IMS resets this value at each synchronization point for the region.

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

CNTN
Displays the name of the region with which the selected region is in unit of work contention.

Type:
Minor of IMS region majors

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

COBA

Displays whether the selected region is the current overflow buffer allocation user.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

DEDC

Displays the number of fast path DEDB calls that are issued during the current invocation of the transaction.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path DEDB, not only fast path (IFP) regions.

DEDR

Displays the number of fast path DEDB reads that are issued during the current invocation of the transaction.

Type:
Minor of RGNF and FSTP

An application DEDB call might require the reading of none or many CIs from DASD to supply the information it requires. One application DEDB call does not necessarily equate to one DEDB CI READ. If the DEDB READ count is greater than the DEDB call count, the application on the average must read multiple CIs to satisfy one DEDB call. If the DEDB READ count is less than the DEDB call count, the application is able to satisfy multiple calls from the same CI.

DEDR.A displays the average number of CI reads per DEDB call.

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path DEDB, not only fast path (IFP) regions.

MSDC

Displays the number of Fast Path MSDB calls issued during the current invocation of the transaction.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path MSDB, not only fast path (IFP) regions.

NBDF

Displays the number of normal buffer allocation (NBA) buffers defined at region initialization.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

NBIU

Displays the number of normal buffer allocation (NBA) buffers this region is using.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

NUOW

Displays the number of Fast Path DEDB units of work contentions that occurred in this region during the current invocation of the transaction.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path DEDB, not only fast path (IFP) regions.

This value is reset at each synchronization point for the region.

OBDF

Displays the number of overflow buffer allocation (OBA) buffers defined at region initialization.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path MSDB, not only fast path (IFP) regions.

IMS resets this value at each synchronization point for the region.

OBIU

Displays the number of overflow buffer allocation (OBA) buffers currently in use by this region.

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that has an OBA specification, not only fast path (IFP) regions.

XCRB

Displays the active DEDB resource (control interval) requests in the region (as represented by the XCRB control block).

Type:
Minor of RGNF and FSTP

This command applies to any region (BMP, MPP, or IFP) that accesses a fast path database, not only fast path (IFP) regions.

If the major command shows more than one region, XCRB shows only the first region that you list. The following list explains the XCRB fields:

dbname
Name of the fast path DEDB.

AREAname
Name of this partition of the DEDB.

RBA
Relative byte address - the address of the resource (control interval).

Status
Status of the resource request. The possibilities are as follows:

EX/OWNER
exclusive/owner

EX/WAITING
exclusive/waiting - highlighted

NE/OWNER
non-exclusive/owner

NE/WAITING
non-exclusive/waiting - highlighted

Owner
Owner of resource displays if status is WAITING

PI
: Is the resource control request known to IMS program isolation? YES or NO

Chapter 9. Database Control (DBCTL) threads

To analyze DBCTL threads, use these OMEGAMON for IMS immediate commands: THRS, THRD, THIN, and TTIM.

The THRS command displays thread summary information, including UOW rates for each connected CCTL. The THRD command displays the CCTL ID, Pseudo-Recovery Token, Recovery Token, RGID (PST number), PSB name, and thread status for each thread. The THRD command also provides the thread state, thread status, elapsed time, and thread occupancy. The THIN thread exception command defines the percentage of threads in use thresholds for CCTLs (CICS regions). The TTIM thread exception command defines time thresholds for PSBs that IMS schedules on behalf of CCTLs threads.

In addition to these immediate commands, you can gather information about DBCTL threads, unless otherwise noted, by issuing the IMS region major commands (described in "IMS region major commands" on page 145) and IMS region minor commands (described in "IMS region minor commands" on page 149.)

DBCTL thread data displays

You can use the Workload menu panel in the Classic menu system to display DBCTL thread summaries and thread details.

Specify option W on the Main menu panel to view DBCTL thread summaries and thread details.

- For a DBCTL-only system, specify option D to display DBCTL thread summaries and thread details.
- For an IMS/DC/DB system, specify option F to display DBCTL thread summaries and thread details.

  Selecting this option displays the thread summaries by CICS (THRS command). To display thread details for the selected CICS region, use the zoom key. To view more information that includes DL/I call statistics for an active thread, press the zoom key on the required thread.

DBCTL thread data commands

You can use the THRD and THRS commands to view thread data for connected CCTLs and CICS transactions.

THRD

Displays the CCTL ID, Pseudo-Recovery Token, RGID (PST number), CICS transaction name, PSB name, thread status, elapsed time, and thread occupancy for each thread.

Type:

Immediate

Format:

THRD [CCTLID=*cctlid*]

The THRD command with no parameters displays thread data for all connected CCTLs (CICSs). An optional CCTLID parameter allows the THRD display to be filtered by the specified *cctlid* value.

The following figure shows an example of the THRD command display.

```
 _______________ KOITHRD  VTM     DBCTL    V510./C I91P 01/25/10 21:58:03    B
> Help PF1         Back PF3         Up PF7         Down PF8           Zoom PF11
>_______________________________________________________________________________
>
>              View Thread Data for Selected CCTL (CICS)
>_______________________________________________________________________________
>
 THRD
+
+                                            =========== Unit-of-Work ===========
+  CCTL      Pseudo    Rgn  CICS  PSB      Thread   Thread    Elapsed    Thread
+   ID       R-Token   ID   Tran  Name     State    Status     Time     Occupancy
+ --------  --------- ----- ----- -----    ------  -------- ---------- ----------
+ CICS22RS             26                   Avail    Idle          0µs      1.50%
+ CICS22RS             35   WD80 DFHSAM05  Active   Ex-Drgn    2m 29s     80.00%
+ CICS22RS             30                   Avail    Idle          0µs       .64%
+ CICS22RS             19                   Avail    Idle          0µs       .64%
+ CICS22RS             34   WD82 DFHSAM05  Active   Ex-Drgn   5,580µs      1.29%
+ CICS22RS             32                   Avail    Idle          0µs       .00%
+ CICS22RS             33                   Avail    Idle          0µs       .64%
+ CICS22RS             23   WD82 DFHSAM05  Active   Ex-Drgn   5,853µs      1.29%
+ CICS22RS             28                   Avail    Idle          0µs       .64%
+ CICS22RS             14                   Avail    Idle          0µs       .64%
+ CICS22RS              8                   Avail    Idle          0µs       .00%
+ CICS22RS              2                   Avail    Idle          0µs       .00%
+ CICS22RS             17                   Avail    Idle          0µs       .00%
+ CICS22RS              4                   Avail    Idle          0µs       .00%
+ CICS22RS              9                   Avail    Idle          0µs       .00%
+ CICS22RS              1                   Avail    Idle          0µs       .00%
+ CICS22RS             16                   Avail    Idle          0µs       .00%
+ CICS22RS             13                   Avail    Idle          0µs       .00%
+ CICS22RS             10                   Avail    Idle          0µs       .00%
+ CICS22RS             12                   Avail    Idle          0µs       .00%
+ CICS22RS              6                   Avail    Idle          0µs       .00%
+ CICS22RS              5                   Avail    Idle          0µs       .00%
+ CICS22RS             11                   Avail    Idle          0µs       .00%
+ CICS22RS             15                   Avail    Idle          0µs       .00%
+ CICS22RS              3                   Avail    Idle          0µs       .00%
```

Figure 113. THRD command display

The fields in the THRD display are as follows:

CCTL ID
Displays the CCTL identifier for the DBCTL thread. For CICS, this ID is the VTAM application ID.

Pseudo R-Token
The Pseudo Recovery Token is only applicable for threads that display a state of Indoubt.

Rgn ID
Displays the IMS region identification number.

CICS Tran
Displays the CICS transaction name for the current thread.

PSB Name
Displays the PSB name.

Thread State
Displays the current state of the thread.

These states are as follows:

Active
The thread is actively processing a PSB. The Thread Status provides more information about the processing that the thread is completing.

Avail
The thread is available to schedule a PSB.

Indoubt
The thread failes and indoubt data exists.

Unavail
The thread ends a scheduled PSB and is therefore unavailable to schedule a new PSB.

Thread Status
The thread status provides more information as to the status of an active thread.

The statuses are as follows:

Thread Inactive

Idle
No application is scheduled in this region.

Scheduler Waits

Wt-Intnt
Waiting because of intent conflict.

Wt-DMB
Waiting for DMB to load.

Wt-PSB
Waiting for PSB to load.

Wt-PSBW
Waiting for PSBW workspace.

Wt-Mover
Waiting for block mover.

Wt-Pools
Waiting for pool space.

Wt-Switch
Waiting to be switched over to an alternate system.

IWAITs

Wt-PIenq
Waiting because of program isolation enqueue conflict.

Wt-IRLM
Waiting because of an IRLM lock conflict.

Wt-DL/I
Waiting for DL/I processing.

Wt-DISP
Waiting in IMS dispatcher.

Wt-TERM
Waiting in termination.

IWAIT
Other IWAIT.

Execution States

Ex-DL/I
Active in DL/I processing.

Ex-Init
Active in scheduling initialization

Ex-Term
Active in termination.

Ex-Drgn
Active in IMS dependent region.

Ex-Abend
Active in ABDUMP manager.

Ex-Lum
Active in LU 6.2 manager.

Ex-Open
Active in dependent region open.

Elapsed Time
Displays the elapsed time that the thread is active, which is based on when the PSB schedule is started.

Thread Occupancy
Displays the occupancy percent for the thread (PST). To calculate this number, OMEGAMON for IMS samples the PSTs during every cycle and observes the number of times it finds an active thread (PSB scheduled) for the PST. For the first 120 samples (or about 10 minutes when the interval is five seconds), the data is not statistically significant, therefore, the value --init-- is displayed. After initialization is complete, the percentage represents the occupancy averaged over the last hour.

More detailed information can be displayed for an active thread by using the PNR*nnn* region major command along with the required region minors. See Chapter 8, "IMS regions," on page 145 for a description of the PNR*nnn* command and the region minors that can be used with this command.

THRS

Displays thread summary information, including Unit-of-Work rates for each connected CCTL (CICS region). The number of Active, Available, Unavailable, and Indoubt threads, and percentage of Active threads are displayed, and Unit-of-Work input count and rate and Unit-of-Work processed count and rate.

Type:
Immediate

There are no parameters on this command.

Figure 114 on page 178 provides a sample of thread summary data.

```
 ________________ KOITHRS  VTM     DBCTL    V510./C I91A 01/25/10 21:58:03    B
> Help PF1         Back PF3         Up PF7         Down PF8            Zoom PF11
>_______________________________________________________________________________
>
>              View Thread Summary Data by CCTL (CICS)
>_______________________________________________________________________________
>
 THRS
+
+                                           =========== Unit-of-Work ===========
+  CCTL     Active  Avail  Unavl  Indbt  Active   Input  Process  Input  Process
+   ID      Count   Count  Count  Count  Percnt   Count   Count   Rate   Rate
+ --------  ------  -----  -----  -----  ------  ------- -------  -----  -----
+ CICS22RS       1     24      0      0    4.00      517     516  10.02  10.00
+ CICS22IP       0     20      0      0     .00        2       2    .00    .00
```

Figure 114. View Thread Summary Data by CCTL (CICS) panel

DBCTL thread exception commands

You can use the THIN and TTIM commands to display and define thresholds for CCTLs.

THIN
Defines the percentage of threads in use thresholds for CCTLs.

Type:
Immediate

Format:
THIN ADD cccccccc THREADS=nnn

ADD
Adds a threshold for a specific CCTL ID.

DELETE
Deletes a threshold for a specific CCTL ID.

LIST
Lists thresholds for specified CCTL IDs.

(blank)
By default, lists all defined thresholds.

cccccccc
Specifies the CCTL ID to which the threshold applies. You can use a wildcard (*) as either the first or last character of the CCTL ID only with the LIST operand.

You can use the THRD immediate command to display all connected CCTL IDs.

THREADS=
Specifies the percentage of threads in use (1-100), that causes the Tnnn exception to trip.

The THIN immediate command assigns a number to each CCTLID that you assign a THREADS= threshold. This number becomes the last three characters of the Tnnn exception name. If a CCTL exceeds the percentage of threads in use threshold, the Tnnn exception trips to notify you that the CCTL has a high percentage of threads in use.

Note: Exception Analysis must be active for the Tnnn exception to provide data.

You can define up to 100 CCTL IDs for the Tnnn exception to monitor during your current session. To save the thresholds that you define with the THIN command in a user profile, use the PPRF command. If you want to create and save more than 100 thresholds, you can use a second user profile.

Figure 115 on page 179 shows an example of adding a threshold for a specific CCTL ID.

```
THIN ADD RCICS321 THREADS=5
+     CCTL THRESHOLD HAS BEEN ADDED
```

Figure 115. THIN Add command

Figure 116 on page 179 shows the list of all defined thresholds for RCI*, after you add the threshold in the previous example.

```
THIN LIST RCI*
+   T001  RCICS001  THREADS=10
+   T002  RCICS002  THREADS=100
+   T003  RCICS003  THREADS=20
+   T004  RCICS004  THREADS=10
+   T005  RCICS321  THREADS=5
+   T006  RCICS322  THREADS=9
```

Figure 116. THIN List command

Figure 117 on page 179 shows an example of deleting a threshold for a CCTL ID.

```
 THIN DEL RCICS322
+     CCTL THRESHOLD HAS BEEN DELETED
```

Figure 117. THIN Delete command

TTIM
Defines time thresholds for PSBs that IMS schedules on behalf of CCTLs threads.

Type:
Immediate

Format:
TTIM ADD cccccccc PSB=aaaaaaaa CPU=nnnn ELAPSED=nnn

ADD
Adds a threshold for a CCTL and a PSB.

DELETE
Deletes a threshold for a CCTL and a PSB.

LIST
Lists thresholds for specified CCTLs.

(blank)
By default, lists all defined thresholds.

The list of defined thresholds sorts in descending alphabetical order, with the most specific threshold at the beginning of the list and thresholds that begin with a wildcard (*) at the end.

cccccccc
Specifies the CCTL ID to which the PSBs and threshold apply. You can use a wildcard (*) as either the first or last character of the CCTL ID only with the LIST operand.

aaaaaaaa
Defines the PSB set to which CPU= and ELAPSED= apply. You can use a wildcard (*) as either the first or last character.

CPU=
Specifies the threshold for processor time, in tenths of a second (1-9999).

ELAPSED=
Specifies the threshold for ELAPSED time, in tenths of a second (1-9999).

You must specify either processor time, ELAPSED time, or both times.

The `TTIM` immediate command compares each active thread with each TTIM threshold to see whether a PSB exceeded the processor or ELAPSED threshold. If a PSB exceeds the processor or ELAPSED time threshold, the TPSB exception trips to notify you that the PSB might be using more processor or ELAPSED time than you expect.

Note: Exception Analysis must be active for the TPSB exception to provide data. If you start OMEGAMON after IMS schedules a PSB, the `TTIM` command does not check the PSB against the TTIM thresholds.

You can define up to 100 PSB names for the TPSB exception to monitor during your current session. To save the PSB names that you define with TTIM in a user profile, use the PPRF command. If you want to create and save more than 100 PSB names, you can use a second user profile.

Figure 118 on page 180 shows an example of adding a threshold for a CCTL and a PSB.

```
 TTIM ADD RCICS410 PSB=ABC* CPU=1 ELAPSED=10
+      PSB THRESHOLD HAS BEEN ADDED
```

Figure 118. TTIM Add command

Figure 119 on page 180 shows the list of all defined thresholds for RCICS410 after you add the threshold in the previous example.

```
 TTIM LIST RCISC410
+      RCICS410 PSB=DFHSAM25 CPU=0001 ELAPSED=0001
+      RCICS410 PSB=DFHSAM2* CPU=0001 ELAPSED=0001
+      RCICS410 PSB=C*       CPU=0001 ELAPSED=0001
+      RCICS410 PSB=B*       CPU=0001 ELAPSED=0001
+      RCICS410 PSB=ABC*     CPU=0001 ELAPSED=0010
+      RCICS410 PSB=*05      CPU=0001 ELAPSED=0001
```

Figure 119. TTIM List command

Figure 120 on page 181 shows an example of deleting a threshold for a CCTL and a PSB.

```
 TTIM DEL RCICS410 PSB=ABC*
+      PSB THRESHOLD HAS BEEN DELETED
```

Figure 120. TTIM Delete command

Chapter 10. IMS resource information

You can issue OMEGAMON for IMS commands to display information about IMS internal and external resources.

The IMS internal and external resources are as follows:

- Scheduling classes
- Data management blocks (DMBs)
- Program specification blocks (PSBs)
- Logical terminals (LTERMs)
- Physical terminals (PTERMs, lines, and VTAM nodes)
- Transactions
- IMS conversations
- IMS pool statistics
- VSAM database pool statistics
- ISAM/OSAM buffer pool statistics
- Fast path resources commands

Scheduling class commands

You can issue scheduling class major commands to select IMS scheduling classes and minor commands to display detailed information about the classes.

The IMS scheduling class is the mechanism by which IMS assigns dispatching priorities to individual transactions. A transaction class is a decimal number from 1 to 25. For this reason, there are no pattern matching major commands for classes.

Scheduling class major commands

Use the scheduling class major commands select IMS scheduling classes.

CLSA
Selects all scheduling classes.

Type:
Major

CLSL
Selects a list of IMS scheduling classes.

Type:
Major

CLSU
Selects IMS scheduling classes that are unusable.

Type:
Major

Use the STAT minor command to display the status of a class, which explains why OMEGAMON for IMS considers the class unusable.

Use the STAT minor command to display the status of a class, which explains why OMEGAMON for IMS considers the class unusable. The following figure is an example of CLSA command output.

```
CLSA         1        2        3        4        5        6
===============================================================================
CLSL         3        2
stat   Started  Started
===============================================================================
CLSU         5
stat   Stopped
```

Figure 121. CLSA major command

The CLSA major command on line 1 in Figure 121 on page 184 shows that the IMS system has six scheduling classes defined. On line 2, the user-selected classes 3 and 2, and then displayed their status in line 3. The CLSU major command tells us that class 5 is unusable and the class is stopped (possibly because a user issued a **/STOP** command).

Scheduling class minor commands

After you issue a scheduling class major command, you can use the following minor commands to display information about the selected IMS scheduling classes.

ARVL

Displays the number of transaction arrivals.

Type:

Minor of scheduling class majors

Format:

ARVLnn

nn indicates that OMEGAMON for IMS considers only transactions of priority *nn* when it calculates the result. If you include an X in the label field (column 1), ARVL excludes any transactions about message-driven BMPs.

If you enter **.R** in the operand field, IMS displays the number of transaction arrivals per second.

The following example shows that 145 class one transactions arrived: 70 of them are priority 4. Over the last OMEGAMON for IMS cycle, the transaction arrival rate for class 1 transactions is 1.2 per second.

```
CLSA         1        2        3
 arvl       145      223       32
 arvl04      70                32
 arvl.R     1.2                .2
```

Figure 122. CLSA command display

IQLN

Displays the input message queue length.

Type:

Minor of scheduling class majors

Format:

IQLNnn

nn indicates that OMEGAMON for IMS considers only transactions of priority *nn* when it calculates the result. If you include an X in the label field (column 1), IQLN excludes any transactions about message-driven BMPs.

A large input queue length might indicate that either the scheduling class is /STOPped or IMS has performance problems and is not scheduling work quickly enough.

The following example shows that class 1 has two input messages queued for class 1, and none of them are priority 4. Class 2 has one priority 4 input message queued and waiting.

PRCS

Displays the number of transactions already processed.

Type:
Minor of scheduling class majors

Format:
PRCSnn

```
CLSA         1        2        3
iqln         2        1
iqln04                1
```

Figure 123. CLSA command display

nn indicates that OMEGAMON for IMSconsiders only transactions of priority *nn* when it calculates the result. If you include an X in the label field (column 1), PRCS excludes any transactions about message-driven BMPs.

If you enter **.R** in the operand field, OMEGAMON for IMS displays the number of transactions per second IMS processed during the last OMEGAMON for IMS cycle.

The following example shows that 145 class one transactions processed; 70 of them were priority 4. Over the last OMEGAMON for IMS cycle the transaction processing rate for class 1 transactions is 1.2 per second.

```
CLSA         1        2        3
prcs       145      223       32
prcs04      70                32
prcs.R     1.2                .2
```

Figure 124. CLSA command display

SMBQ
Displays the number of SMBs enqueued.

Type:
Minor of scheduling class majors

Format:
SMBQnn

nn indicates that OMEGAMON for IMS considers only transactions of priority *nn* when it calculates the result. If you include an X in the label field (column 1), SMBQ excludes any transactions about message-driven BMPs.

This is not the same as IQLN (input message queue length); SMBQ represents the number of different transactions that are currently enqueued, not the number of times the user issued the transaction.

The following example shows that two different transactions are queued and waiting for class one scheduling; one of them is priority 4. While there are only two transaction types queued, 45 input messages are waiting.

```
CLSA         1        2        3
smbq         2                 2
smbq04       1
iqln        45                 3
```

Figure 125. CLSA command display

STAT
Displays the status of this scheduling class.

Type:
Minor of scheduling class majors

The possibilities are as follows :

stopped
IMS received a */STOP* CLASS command.

active
A transaction of this class is running.

started
The class is schedulable but inactive.

The following example shows the status of each scheduling class.

```
CLSA         1        2         3
stat   Started  Stopped   Active
```

Database management block commands

The database management block (DMB) is an internal IMS control block that defines an IMS database. IMS identifies each database by an eight-character name that you choose during the IMS system generation process in the DBO= parameter of the database macro.

- "DMB immediate commands" on page 186
- "DMB major commands" on page 187
- "DMB minor commands" on page 189

DMB immediate commands

You can use the DBIO immediate command to display information about IMS databases, which includes the database name, type, and status.

DBIO
Displays each IMS database and associated DDName with current I/O rates and EXCP counts. This information is not available for fast path databases.

Type:
Immediate

You can filter the display by setting a variable. The variables available are as follows:

OIDBTYP
Database type

Maximum of eight characters

Valid values are HDAM, HIDAM, HISAM, HSAM, SSAM, DEDB, MSDB, INDEX

OIDBORG
Database organization

Maximum of four characters

Valid values are OSAM, VSAM

OIDBPAT
Database name pattern

Maximum of 8 characters. Use an asterisk for wild characters. If an asterisk is followed by a space, the remainder of the eight-character name is wild.

OIDBST
DMB status

Maximum of 16 characters

Valid values are

'/DBR ACTIVE'
'/DBD ACTIVE'

’WAIT ON DMB POOL’
’STOP PENDING’
’RESTART PEND.’
’RECOVERY NEEDED’
’DYN ALLOC ERROR’
’DMB STOPPED’
’DMB NOT LOADED’
’DMB NOT FOUND’
’DMB NOT ALLOC’
’DMB LOCKED’
’DMB AVAILABLE’
’DMB ACTIVE’
’DATA AREA EMPTY’
’AREA STOPPED’
’AREA OPEN’
’AREA NOT OPEN’
’AREA I/O ERROR’
’USABLE’
’UNUSABLE’

You must use single quotation marks to enclose the literal because these literal values have embedded spaces. After a variable is set, it remains set during your session or until it is changed. To set a variable, issue the following command that starts in column 1.

```
-VAR S & variable value
```

The variable name must be preceded by an ampersand (&). To disable a variable, set the variable value to asterisk (*). The following figure is an example of a typical DBIO command display,

```
DBIO
         DBD Name   DMB Status      Type/Org    DD Name   EXCP Count   EXCP Rate
         --------   -------------   ----------  -------   ----------   ---------
         BA$HDL00   DMB AVAILABLE   HDAM/OSAM   BAHDL001           0         .00
         BB$HDL00   DMB AVAILABLE   HDAM/OSAM   BBHDL001           0         .00
         BC$HDL00   DMB AVAILABLE   HDAM/OSAM   BCHDL001           0         .00
         BD$HDM00   DMB AVAILABLE   HDAM/OSAM   BDHDM001           0         .00
         BE$HDM00   DMB AVAILABLE   HDAM/OSAM   BEHDM001           0         .00
         BF$HDM00   DMB AVAILABLE   HDAM/OSAM   BFHDM001           0         .00
         BG$HDH00   DMB AVAILABLE   HDAM/OSAM   BGHDH001           0         .00
         BG$HDH00   DMB AVAILABLE   HDAM/OSAM   BGHDH001           0         .00
         BH$HDH00   DMB AVAILABLE   HDAM/OSAM   BHHDH001           0         .00
         BI$HDH00   DMB AVAILABLE   HDAM/OSAM   BIHDH001           0         .00
         DBFSAMD1   DMB AVAILABLE   MSDB                           0         .00
         DBFSAMD2   DMB AVAILABLE   MSDB                           0         .00
         DBFSAMD4   DMB ACTIVE      HDAM/VSAM   LOAN               0         .00
         DI21PART   DMB AVAILABLE   HISAM/VSAM  DI21PART           7         .00
         DI21PART   DMB AVAILABLE   HISAM/VSAM  DI21PARO           3         .00
         IVPDB1     DMB NOT LOADED                                 0         .00
         IVPDB1I    DMB NOT LOADED                                 0         .00
         IVPDB2     DMB NOT LOADED                                 0         .00
         IVPDB4     DMB AVAILABLE   MSDB                           0         .00
         DBFSAMD3   AREA NOT OPEN   DEDB                           0         .00
         IVPDB3     AREA OPEN       DEDB        DFSIVD32           0         .00
         IVPDB3     AREA OPEN       DEDB        DFSIVD34           0         .00
```

Figure 126. DBIO command display

DMB major commands

You can use the DMB major commands to select database management blocks (DMBs).

DMBA
Selects all DMBs.

Type:
Major

DMBE
Selects all DMBs that have I/O or dynamic back-out errors.

Type:
Major

I/O errors consist of read errors, write errors, hot I/O standby errors, DBRC user errors, and DBRC permanent errors.

DMBI
Selects DMBs that are currently loaded into virtual storage.

Type:
Major

IMS loads only DMBs that it has marks resident into memory or when the database they represent is open.

DMBL
Selects all specified DMBs.

Type:
Major

DMBN
Selects databases that are defined as unrecoverable.

Type:
Major

The DMBN major command displays only unrecoverable databases that are opened.

DMBP/n
Selects all DMBs that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If DMBP does not find a /n value in the argument field, it uses the first pattern that is provided with the last **.SPT** command.

DMBU
Selects DMBs that are unusable.

Type:
Major

Use the SCHD minor command to display the status of a DMB, which explains why IMS considers a DMB unusable.

In some IMS systems, IMS does not find multiple DMBs at IMS initialization and OMEGAMON for IMS includes them in the unusable display. At times, you might not care that these DMBs are unavailable; to exclude them from the DMBU display, enter an *X* in the argument field (DMBUX).

The following example shows the database management block major commands.

```
 DMBA  BE2PCUST BE3ORDER BE3ORDRX BE3PARTS BE3PSID1 DI21PART
 schd  Availble   Active  Not-Fnd Availble Availble DMB-Stop
================================================================================
 DMBI  BE3ORDER
================================================================================
 DMBL  BE3ORDER BE3ORDRX
================================================================================
 .SPT  B**P*     <--- Enter pattern - Current pattern = ‘B**P*’
 DMBP  BE2PCUST BE3PARTS BE3PSID1
================================================================================
 DMBU  BE3ORDRX DI21PART
 schd   Not-Fnd DMB-Stop
```

Figure 127. Database management block major commands

The **DMBA** major command on line 1 of the previous example shows that the IMS system has six defined DMBs. The **SCHD** minor command on line 2 displays the scheduling status of each DMB. The **DMBI** major command selected the only DMB currently in memory (since it is the only one currently active). The **DMBL** major selected the two DMBs the user-specified list requested. The **DMBP** major selects those DMBs whose names match the pattern which the previous **.SPT** immediate command set. The **DMBU** command selects the only two DMBs, which are unschedulable, and the **SCHD** minor shows the reason for each.

DMB minor commands

You can use the DMB minor commands to display more information about database management blocks (DMBs).

ACCS
Displays the highest database access intent that is defined for the resident (or currently open) DMB.

Type:
Major

ACCS can display the following access intents:

Rd-Only
Read-only access

Read
Read access

Update
Update access

Xclusive
Exclusive access

Unknown?
Any other type displays as unknown

If the DMB is not currently in virtual memory, **--------** displays.

ADMB
Displays the address of the DMB block in storage.

Type:
Minor of DMB majors

If the block is not currently in storage, **--------** displays.

CASP
Displays the number of control area splits that occur for a VSAM database.

Type:
Minor of DMB majors

The sum of control area (CA) splits for all VSAM data sets that make up the database since the last time VSAM reorganized the database. Before a database opens, the display reads Not-Open. If the

DMB does not represent a VSAM database, Not-VSAM displays. If the DMB represents a fast path database, FASTPATH displays.

CISP
Displays the number of control interval splits that occurred for a VSAM database.

Type:
Minor of DMB majors

The sum of control interval (CI) splits for all VSAM data sets that make up the database since the last time VSAM reorganized the database. Before a database opens, the display reads Not-Open. If the DMB does not represent a VSAM database, Not-VSAM displays. If the DMB represents a fast path database, FASTPATH displays.

DRES
Displays the residency status of the DMB.

Type:
Minor of DMB majors

The DMB status displays as Resident, In-memry, or Not-in. In-memry implies the DMB is in memory but IMS does not mark it resident.

DSZE
Displays the size of a DMB in decimal bytes.

Type:
Minor of DMB majors

This number must be the same as the number on an ACBGEN listing.

ID#
Displays either the DL/I ddname or the fast path area name for databases that incur I/O errors.

Type:
Minor of DMB majors

Format:
ID#nn

ID# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of relative I/O errors).
- The letter *A* (displays all I/O errors).

  The default value is 1.

IR#
Displays the RBA of the I/O error in the database.

Type:
Minor of DMB majors

Format:
IR#nn

IR# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of relative I/O errors).
- The letter A (displays all I/O errors).

The default value is 1.

IT#
Displays the type of I/O error in the database.

Type:
Minor of DMB majors

Format:
IT#nn

IT# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of relative I/O errors).
- The letter A (displays all I/O errors).

The default value is 1.

IT# displays the following error types:

RD-err
DMB has a read error.

WRT-err
DMB has a write error.

IOT-err
DMB has a hot standby error.

DBRC-err
DMB has a DBRC user error.

PRM-err
DMB has a DBRC permanent error.

ODDN
Displays information about non-VSAM databases.

Type:
Minor of DMB majors

ODDN displays the following information about non-VSAM databases: DDname, blocksize, volume serial number (volser), and unit address of the device on which the database is stored. If the database is on multiple volumes, the volser and unit address of each volume displays.

PB#
Displays the PSB of the database that has a dynamic back-out error.

Type:
Minor of DMB majors

Format:
PB#nn

PB# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of relative dynamic back-out errors).
- The letter A (displays all dynamic back-out errors).

The default value is 1.

PD#
Displays the Julian date when dynamic back-out error occurred against the database.

Type:
Minor of DMB majors

Format:
PD#nn

PD# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of relative dynamic back-out errors).
- The letter *A* (displays all dynamic back-out errors).

The default value is 1.

PSBC
Displays the number of PSBs that are currently active and using this DMB.

Type:
Minor of DMB majors

PT#
Displays the time when the dynamic back-out error occurred against the database.

Type:
Minor of DMB majors

Format:
PT#nn

PT# accepts the following values in its argument field:

- Any number in the range 1-99 (represents the number of the relative dynamic back-out errors).
- The letter *A* (displays all dynamic back-out errors).

The default value is 1.

The time is in the format *hh*:*mm*:*ss*

RCOV
Displays whether a database is recoverable or unrecoverable.

Type:
Minor of DMB majors

RCOV displays the following information:

Not-Fnd
DMB cannot be found at IMS startup and the database is not usable.

Non-Rcov
The database is opened and unrecoverable.

Not-Opnd
The database is not opened. RCOV cannot determine the status.

Recovrbl
The database is recoverable.

RCOV displays unrecoverable databases only when the databases are opened.

SCHD
Displays the current scheduling status for this DMB.

Type:
Minor of DMB majors

The following list explains the possible messages:

/DBD-Act
A **/DBD** command is active, and the DMB is not available.

DMB-Stop
A user issued a **/STOP DATABASE** command against the DMB.

DMB-Lock
A user issued a **/LOCK DATABASE** command against the DMB.

Not-Fnd
A user defined the DMB but IMS does not find it in ACBLIB at startup.

WF-Pool
The DMB is coming into memory, but is waiting for DMB pool space.

Active
A PSB that uses the DMB is scheduled (DMB is not necessarily open).

Availble
The database is available but not currently in use.

SPD#
Displays the subpool ID of each OSAM data set in the database.

Type:
Minor of DMB majors

Format:
SPD#nn

The variable *nn* specifies an OSAM data set number. This number is required. If the data set is not assigned to a specific subpool, SPD# displays **default**

STAT
Displays the scheduling status of the database.

Type:
Minor of DMB majors

STAT is an alias of the SCHD minor command.

TYPE
Displays the database type.

Type:
Minor of DMB majors

The following list explains the possible output:

VSAM
The database is stored as a VSAM file.

OSAM
The database is stored as an OSAM file.

DEDB
The database is a fast path DEDB.

MSDB
The database is a fast path MSDB.

Not in
The DMB control block is not currently in memory. Therefore, this information is not available.

VDDN
Displays information about VSAM databases.

Type:
Minor of DMB majors

VDDN displays the following information:

- ddnames that make up the database
- Number of CA and CI splits
- Number of data set extents
- Whether VSAM writecheck is on or off
- CI-size of the data component
- CI-size of the index component (if any)
- Volume serial number
- Unit address of the device on which the data set is stored

If the database is on multiple volumes, VDDN displays the volser and unit address of each volume.

OMEGAMON for IMS produces a display only for the first major item that you select.

If the DMB does not represent a VSAM database, VDDN displays

```
>> Not a VSAM Database >>
```

If the database that DMB represents is not open, VDDN displays

```
>>Database not open<<
```

Program specification block commands

You can issue OMEGAMON for IMS program specification block (PSB) major commands to select PSBs and minor commands to display more information about the selected PSBs.

The *program specification block (PSB)* is an internal IMS control block that defines the application programs in an installation. IMS identifies each PSB by an eight-character name that you choose during the IMS system generation process in the PSB= parameter of the APPLCTN macro.

PSB major commands

Use a PSB major command to select PSBs.

PSBA
Selects all PSBs.

Type:
Major

PSBI
Selects all PSBs that are currently loaded into virtual storage.

Type:
Major

IMS loads only non-resident PSBs (those PSBs that the user does not mark resident at IMS gen time) into memory when IMS schedules the program they represent. The PSBI major selects all those PSBs that are currently loaded into virtual storage, which includes all resident PSBs and any PSBs that might be loaded into the PSB pool even though they are not currently scheduled.

PSBL
Selects all specified PSBs.

Type:
Major

PSBP/n
Selects all PSBs that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If PSBP does not find a /n value, it uses the first pattern that is provided with the last **.SPT** command.

PSBS
Selects all PSBs that are currently running in at least one dependent region.

Type:
Major

These are a subset of PSBs that the PSBI major command selects.

PSBU
Selects all PSBs that are unusable.

Type:
Major

Use the SCHD minor command to display the status of a PSB, which explains why IMS considers a PSB unusable. (See "PSB minor commands" on page 195 for details.)

In some IMS systems, IMS does not find multiple PSBs at IMS initialization. OMEGAMON for IMS includes these PSBs in the unusable display. Sometimes, you might not care that these PSBs are unavailable. To exclude PSBs from the PSBU display, enter an X in the argument field (PSBUX).

PSB minor commands

Use the PSB minor commands to display more information about selected program specification blocks (PSBs).

APSB
Displays the address of the PSB if the block is in storage.

Type:
Minor of PSB majors

NotInMem displays if the block is not in storage.

ARVL
Displays the number of transaction arrivals that are associated with this PSB.

Type:
Minor of PSB majors

Use this command with the **.R** suffix to display the rate of transactions arrivals.

DB#
Displays DMBs associated with a selected PSB.

Type:
Minor of PSB majors

Format:
DB#nnn

You can start the DB# minor command in two different ways.

1. To request a multi-line display of all the DMBs associated with the first PSB selected, enter an A in the label field without *nn*.

```
PSBL  PE4CPPUR PE4CODEL
Adb#   Database   Highest Intent
+      BE3PSID1   Update
+      BE3PARTS   Update
+      BE3ORDER   No Intent
+      BE3ORDRX   No Intent
```

Figure 128. DB# command display

In the example, PSB PE4CPPUR has two databases that are associated with it, which have a highest intent of update. (DB# does not produce any output for the PE4CODEL PSB.) If the intent list for the PSB is not in memory, the DB# minor produces the message

```
        >> Intent list for PSB PE4CPPUR is not in memory <<.
```

2. To display the name of the *nn*th DMB associated with the PSB, leave the label field blank and supply a number in the argument field.

```
PSBL  PE4CPPUR PE4CODEL
db# 1 BE3PSID1 BE2PCUST
db# 2 BE3PARTS BE3ORDER
db# 3 BE3ORDER BE3PARTS
db# 4 BE3ORDRX BE3ORDRX
db# 5  Max=4   BE3PSID1
```

Figure 129. DB# command display

If the PSB you specify is not in memory, **NotInMem** displays.

IRES
Displays the residency attribute of the intent list that is associated with a PSB.

Type:
Minor of PSB majors

The intent list can either be resident or not resident. If not resident, the intent list can be either in memory or out. IRES displays these three states as Resident, In-memory, and Not-in.

LANG
Displays the program language that is associated with a PSB.

Type:
Minor of PSB majors

The language is either ASM/CBL, Pascal, or PL/1. (If the PSB is not currently in memory the display shows NotInMem.)

PDLS
Displays the number of bytes used in the DPSB pool.

Type:
Minor of PSB majors

When you choose the LSO=S option, each active PSB requires space in the DPSB pool in the DLISAS address space.

If the PSB is not currently in the pool, NOT-IN displays. If you do not choose the LSO=S option, PDLS displays the value **DLS inac.** If all PSBs are in CSA, PDLS displays ALL-CSA.

PDRA
Displays the virtual address of the PSB directory entry (PDIR) associated with this PSB.

Type:
Minor of PSB majors

This address displays in hexadecimal.

PPUS
Displays the amount of PSB pool space that must be set aside for this PSB to load into memory.

Type:
Minor of PSB majors

This address displays in hexadecimal.

The **PSZE** minor command gives the size of the PSB, but the actual amount of space it needs from the pool depends on the residency attributes of the PSB, its intent list, and whether the PSB is to be parallel scheduled. (A resident PSB with a resident intent list that is not parallel scheduled does not require any space from the PSB pool, since IMS pre-assigns all necessary memory from resident pools.)

If the PSB is parallel scheduled, IMS makes a copy of the PDIR in the PSB pool. PPUS displays the result, which includes the size of the PDIR.

To find out how much total PSB pool storage is in use multiply the size PPUS reports by the current number of schedulings of the PSB (see the **PSBC** minor command).

PRCS
Displays the number of transactions that are associated with this PSB that were processed.

Type:
Minor of PSB majors

PRES
Displays the residency status of this PSB.

Type:
Minor of PSB majors

IMS marks a PSB RESIDENT, DOPT, or neither. If not RESIDENT, the PSBn can be either currently in memory or out. PRES displays these states as Resident, DOPT-in, DOPT-out, In-memory, and Not-in. This has nothing to do with whether IMS schedules the PSB, since RESIDENT PSBs are always in memory, and non-RESIDENT PSBs can remain in the PSB pool even if they are not currently in use.

PSBC
Displays the number of times the PSB is scheduled.

Type:
Minor of PSB majors

For a PSB that cannot be parallel scheduled, this number cannot be greater than 1.

Zero means that the PSB is not currently scheduled.

PSZE
Displays the size of the PSB in decimal bytes.

Type:
Minor of PSB majors

This number must be the same as that found on an ACBGEN listing.

PTYP
Displays the program type that is associated with this PSB.

Type:
Minor of PSB majors

Program type is either online or batch.

SCHD
Displays the current scheduling status for this PSB.

Type:
Minor of PSB majors

SCHD displays the current scheduling status for this PSB; the following list explains the possible messages:

DMB-Stop
The user issued a **`/STOP DATABASE`** command against one of the PSB's databases.

PSB-Lock
The user issued a**`/LOCK PROGRAM`** command against the PSB.

PSB-Stop
The user issued a **`/STOP PROGRAM`** command against the PSB or the PSB abended.

Not-Fnd
The PSB is defined but not found in ACBLIB at IMS startup.

Schduled
The PSB is scheduled at least once.

Schdlble
The PSB is usable but not currently in use.

STAT
Displays the status of the PSB.

Type:
Minor of PSB majors

STAT is an alias of the SCHD minor command.

STYP
Displays the type of scheduling that is associated with this PSB.

Type:
Minor of PSB majors

Scheduling type is either parallel or serial.

TRCE
Displays the status of the trace option for this PSB.

Type:
Minor of PSB majors

Trace option status is one of the following states: off (Off), on (Trace On), or on with the compare option (TraceCmp).

TX#
Displays the transactions that are associated with the PSB.

Type:
Minor of PSB majors

Format:
TX#nnn

To request a multi-line display showing all the transactions that are associated with the first PSB selected, enter an A in the label field.

The following example shows that PSB DFSSAM04 has four transactions that use it. (ATX# does not produce any output for the DFSSAM05 PSB.)

```
 PSBL  DFSSAM04 DFSSAM05
Atx#   Transaction
+      ADDINV
+      ADDPART
+      DLETINV
+      DLETPART
```

Figure 130. ATX# command display

Leave the label field blank and supply a number in the argument field to display the name of the *nn*th transaction that is associated with the PSBs.

```
 PSBL  DFSSAM04 DFSSAM05
 tx# 1   ADDINV    CLOSE
 tx# 2  ADDPART    Max=1
 tx# 3  DLETINV    Max=1
 tx# 4 DLETPART    Max=1
```

Figure 131. TX# command display

If there is no transaction corresponding to the number you supply, **Max=nnn** displays. Batch programs have no transactions that are defined at all, and respond with (Batch).

IMS user commands

You can use a number of OMEGAMON for IMS commands to select and display information about IMS users.

IMS user major commands

Issue the IMS user major commands to select IMS users.

USRA
Selects all users signed on to a node or allocated to a node.

Type:
Major

USRC
Selects all users in an active conversation.

Type:
Major

USRD
Selects all users with dead letter queues.

Type:
Major

USRE
Selects all users in EXCLUSIVE mode.

Type:
Major

USRI
Selects all users in RESPONSE mode with IN-DOUBT status.

Type:
Major

USRL
Selects all users in PRESET mode.

Type:
Major

USRM
Selects all users in MFSTEST mode.

Type:
Major

USRO
Selects all users in a held conversation.

Type:
Major

USRP/n
Selects all users that match a pattern. The pattern is set with the **.SPT** command.

Type:
Major

USRR
Selects all users in RESPONSE mode.

Type:
Major

USRS
Selects all users that are statically defined through system definition.

Type:
Major

USRT
Selects all stopped users.

Type:
Major

USRZ
Selects all users in TEST mode.

Type:
Major

IMS user minor commands

Issue the IMS user minor commands to display more information about IMS users.

USID
The RACF userid that is used to sign on (N/A if not associated with node)

Type:
Minor of IMS user major commands

NODE
The Node to which the user is signed on is assigned (N/A if not associated with node)

Type:
Minor of IMS user major commands

ENQ
The cumulative count of items that are enqueued for this user

Type:
Minor of IMS user major commands

DEQ
The cumulative count of items that are dequeued for this user

Type:
Minor of IMS user major commands

QUE
The count of items still on the queue for this user

Type:
Minor of IMS user major commands

STAT
Displays status of the user.

Type:
Minor of IMS user major commands

The status of the user is one of the following states:

Alloc
User is signed on or allocated.

Conv-Act
User is in an active conversation.

Conv-Hld
User is in a held conversation.

Deadq
User has dead letter queue.

Excl
User is in EXCLUSIVE mode.

Mfstest
User is in MFSTEST mode.

Preset
User is in PRESET mode.

Resp
User is in RESPONSE mode.

Resp-Inp
User is in RESPONSE mode with IN-DOUBT status.

Static
User is statically defined.

Stopped
User is stopped.

Test
User is in Test mode.

Logical terminal commands

You can use a number of OMEGAMON for IMS commands to select and display information about logical terminals (LTERMs).

Logical terminal major commands

Use the logical terminal major commands to select logical terminals (LTERMs).

TRMA
Selects all logical terminals.

Type:
Major

TRMG
Selects logical terminals in group *nn*.

Type:
Major

Format
TRMGnn

RTA and DEXAN also monitor these groups.

TRML
Selects all specified logical terminals.

Type:
Major

TRMP/n
Selects all logical terminals that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns.

If you do not specify a value for /n, TRMP uses the first pattern set with the **.SPT** command (pattern 0).

TRMQ
Selects those logical terminals with an output message queue length greater than or equal to *nn*

Type:
Major

Format:
TRMQnn

TRMS
Selects LTERMs associated with an IMS VTAM node physically connected to IMS.

Type:
Major

TRMU
Selects logical terminals that are unusable.

Type:
Major

TRPQ

Selects logical terminals whose names match a pattern and that have an output message queue length >= *nn.*

Type:

Major

Format

TRPQnn

nn

Specifies the output message queue length.

You can set the pattern with the **.SPT** command. TRPQ uses only the first pattern (pattern 0).

Logical terminal minor commands

Use the logical terminal minor commands to display more information about logical terminals (LTERMs).

ATOL

Displays the average length of the output messages that were sent to this LTERM.

Type:

Minor of logical terminal majors

The length is in bytes.

OMEGAMON for IMS does not count messages still in the message queue. For example, if there are 10 messages and three are in the message queue, ATOL displays only the average lengths of the remaining seven.

LNID

Displays the line ID number that is associated with an LTERM.

Type:

Minor of logical terminal majors

Each BTAM terminal has its own line number. This number reflects the order in which the IMS system definition specified the terminals.

LTRX

Displays the name of the last transaction that is issued from this LTERM.

Type:

Minor of logical terminal majors

The LTERM of the target terminal displays if the last action is a message switch to send a message to another terminal.

This field is set to blanks for a terminal that does not issue a transaction or, which is not logged on to IMS. IMS loses the last transaction that is issued when the terminal user issues any MTO command (that is, an IMS command).

MDEQ

Displays the number of messages that were dequeued from the LTERM.

Type:

Minor of logical terminal majors

This indicates the number of output messages IMS displayed at the LTERM.

MENQ

Displays the number of messages that were enqueued to the LTERM.

Type:

Minor of logical terminal majors

This means that the number of output messages IMS enqueued to the LTERM.

NODE
Displays the VTAM node name that is associated with the LTERM.

Type:
Minor of logical terminal majors

If the LTERM does not have a VTAM node (BTAM terminals, for example) **--none--** displays.

OQLN
Displays the current output queue length for the LTERM.

Type:
Minor of logical terminal majors

The number of messages IMS enqueued to the LTERM but is not yet dequeued. This queue length is equal to the total number of messages enqueued (MENQ) minus the total number of messages dequeued (MDEQ). If this number is greater than zero, it means something is preventing IMS from physically delivering the message to the LTERM (such as the terminal is /STOPped).

PTID
Displays the PTERM ID associated with an LTERM.

Type:
Minor of logical terminal majors

SNID
Displays the user sign-on ID associated with an LTERM.

Type:
Minor of logical terminal majors

-None- means not signed on. **-n/a-** means no IMS security.

STAT
Displays the status of the LTERM.

Type:
Minor of logical terminal majors

The status is one of the following states:

Stopped
A **`/STOP LTERM`** command is issued against the LTERM.

Pstopped
A **`/PSTOP LTERM`** command is issued against the LTERM.

Purged
A **`/PURGE LTERM`** command is issued against the LTERM.

Locked
A **`/LOCK LTERM`** command is issued against the LTERM.

Started
The LTERM is available for use.

If the LTERM is in any state other than started,OMEGAMON for IMS considers it unusable and TRMU selects it.

Physical terminal and line commands

You can use a number of OMEGAMON for IMS commands to select and display information about physical terminals (PTERMs) and lines.

- "Physical terminal major commands" on page 204
- "Physical terminal minor commands" on page 204
- "IMS line major commands" on page 205
- "IMS line minor commands" on page 205

Physical terminal major commands

Use the physical terminal major commands to select IMS physical terminals (PTERMs).

A PTERM is not a VTAM node, but the physical ordering of terminals within IMS. For example, PTERM L0005T01 represents terminal one on line five. The device might not have a VTAM node that is associated with it.

The physical terminal major commands select only BTAM terminals.

PTRA
Selects all IMS physical terminals.

Type:
Major

PTRL
Selects all specified IMS physical terminals.

Type:
Major

PTRP/n
Selects all PTERMs that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If PTRP does not find a /n value, it uses the first pattern supplied with the last **.SPT** command.

PTRU
Selects all unusable IMS physical terminals.

Type:
Major

Physical terminal minor commands

Use the physical terminal minor commands to display detailed information about selected physical terminals (PTERMs).

DETL
Provides detailed information about the first terminal displayed.

Type:
Minor of physical terminal majors

For a deeper understanding of this information, see the IMS CTB data area description.

ICNT
Displays the number of input messages from this PTERM.

Type:
Minor of physical terminal majors

LINE
Displays the relative line number of this PTERM.

Type:
Minor of physical terminal majors

OCNT
Displays the number of output messages from this PTERM.

Type:
Minor of physical terminal majors

RTRM
Displays the relative number of this PTERM within its line.

Type:
Minor of physical terminal majors

STAT
Displays the status of this PTERM.

Type:
Minor of physical terminal majors

The status of this line is one of the following states:

started
The PTERM is active.

not-oper
The PTERM is not operable (it is turned off or offline).

locked
The PTERM is locked.

stop-ioq
The PTERM is stopped for input (i), output (o), or queuing (q).

IMS line major commands

Use the IMS line major commands to select IMS lines. The commands select only BTAM terminals.

LNEA
Selects all IMS physical terminal lines.

Type:
Major

LNEI
Selects all idle IMS physical terminal lines.

Type:
Major

LNEL
Selects all specified IMS physical terminal lines.

Type:
Major

LNEP/n
Selects all IMS physical terminal lines that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If LNEP does not find a /n value, it uses the first pattern supplied with the last **.SPT** command.

LNEU
Selects all unusable IMS physical terminal lines.

Type:
Major

IMS line minor commands

Use the IMS line minor commands to display information about selected IMS lines.

NODE
Displays the VTAM node name.

Type:
Minor of IMS line majors

OMEGAMON for IMS displays **--none--** if this is a BTAM line.

STAT
Displays the status of this line.

Type:
Minor of IMS line majors

The status of this line is one of the following states:

started
The line is active.

idle
The line is idle.

stop
The line is stopped.

stop-IO
The line is stopped for input or output.

Figure 132 on page 206 shows the output of the LNEA major command with the NODE and STAT minor commands.

```
LNEA     L0001    L0002    L0003    L0004    L0005    L0006    L0007    L0008  +
node   --none-- --none-- --none-- --none-- --none-- --none-- --none-- --none--
stat       Idle  Stop-IO     Idle     Idle     Idle     Idle     Idle     Idle
```

Figure 132. Physical line minor commands

IMS VTAM node commands

You can issue OMEGAMON for IMS commands to select IMS VTAM nodes and display detailed information about the selected nodes.

IMS VTAM node major commands

Issue the `NODx` commands to select IMS VTAM nodes.

NODA
Selects all IMS VTAM nodes, including LU 6.2 resources.

Type:
Major

NODL
Selects all specified IMS VTAM nodes, including LU 6.2 resources.

Type:
Major

NODP/n
Selects all IMS VTAM nodes that match a pattern, including LU 6.2 resources.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If NODP does not find a /n value, it uses the first pattern that is provided with the last **.SPT** command.

NODQ
Selects all IMS VTAM nodes whose output queue length is equal to or greater than *nn* 1, including LU 6.2 resources.

Type:
Major

Format:
NODQnn

NODS
Selects all IMS VTAM nodes that are physically connected to IMS, including LU 6.2 resources.

Type:
Major

Format:
cNODS

Where *c* can be:

blank
Selects all VTAM nodes that are connected to IMS.

I
Selects all VTAM nodes that are connected to IMS that are active in input.

O
Selects all VTAM nodes that are connected to IMS that are active in output.

IMS VTAM node minor commands

Issue the following minor commands to display information about selected IMS VTAM nodes.

CID
Displays the communication ID (CID).

Type:
Minor of IMS VTAM node majors

CTB
Displays the CTB address that is associated with this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

INCT
Displays the total number of messages that are received by this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

MDEQ
Displays the number of messages that are dequeued at this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

MENQ
Displays the number of messages that are enqueued to this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

OQLN
Displays the current output queue length for this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

STAT
Displays the status of this IMS VTAM node.

Type:
Minor of IMS VTAM node majors

Possible statuses are as follows:

C1
Component 1 is inoperable.

C2
Component 2 is inoperable.

C3
Component 3 is inoperable.

C4
Component 4 is inoperable.

CA
Node is in an active conversation.

CD
CLSDST is done for this node.

CH
Node is in a held conversation.

CO
Node is physically connected to IMS.

DE
Node is deactivated.

EX
Node is in exclusive mode.

FO
Session initiation option of FORCE is specified.

ID
Node is idle.

IN
Node is inoperable.

LK
Node is locked.

LO
The losterm exit is driven for this node.

MT
Node is in MFS test mode.

OD
OPDNST is pending for this node.

QU
Node is quiesced.

RE
Node in response mode.

RI
Node is in response mode, in-doubt.

RS
Node is in resync mode.

SH
Node is in shut mode.

SI
Signon is complete for this node.

SL
SIMLOGON

ST
Node is stopped.

TE
Node is in test mode.

TR
Node is traced.

TYPE
Displays the terminal type.

Type:
Minor of IMS VTAM node majors

TYPE displays one of the following terminal types:

3286 FIN (financial device) 3614 3277 LU6.2 LUT6 3790 SLU1 SLU2 SLU4 NTO SLUP

Transaction commands

You can issue a number of OMEGAMON for IMS commands to select and display information about transactions.

Transaction major commands

Use the transaction major commands to select transactions.

TRXA
Selects all transactions.

Type:
Major

TRXC
Selects all transactions that belong to scheduling class ***nn.***

Type:
Major

Format:
TRXCnn

To select three-digit scheduling classes, use the TXCnnn command.

TRXD
Selects all transactions that run in a dependent region.

Type:
Major

TRXG
Selects all transactions that belong to transaction group *nn*.

Type:
Major

Format:
TRXGnn

TRXL
Selects all specified transactions.

Type:
Major

TRXP/n
Selects all transactions that match a pattern.

Type:
Major

The **.SPT** command sets the pattern. See the **.SPT** command for information about setting patterns. If TRXP does not find a /n value, it uses the first pattern that is supplied with the last **.SPT** command.

TRXQ
Selects those transactions with an input queue length greater than or equal to *nn*.

Type:
Major

Format:
TRXQnn

Any transactions with at least *nn* transactions that IMS does not process yet (perhaps because there is no message region available in which to run them) are selected.

TRXS
Selects all transactions that have messages that are enqueued on their suspend queue and, which are USTOPPED.

Type:
Major

All the current minors for the TRXc major commands apply to the TRXS major command.

TRXU
Selects all transactions that are unschedulable.

Type:
Major

The transaction cannot be scheduled because it is in one of the following states:

Locked
A **/LOCK TRAN** command is issued against the transaction.

Norgns
No region is assigned for this transaction.

Pstopped
A**/PSTOP TRAN** command is issued against the transaction.

Purged
A**/PURGE TRAN** command is issued against the transaction.

Stopped
A **/STOP TRAN** command is issued against the transaction.

Ustopped
No longer available for processing.

TXC
Selects all transactions that belong to scheduling class *nnn*.

Type:
Major

Format:
TXCnnn

To select two-digit scheduling classes, use the TRXCnn command.

TXPQ
Selects all transactions whose names match a pattern and whose queue lengths are greater than or equal to *nn.*

Type:
Major

Format:
TXPQnn

Transaction minor commands

Issue the transaction minor commands to display detailed information about transactions.

ARVL
Displays the number of transaction arrivals for this transaction.

Type:
Minor of transaction majors

For transactions that *must* execute in a Fast Path (IFP) region, ARVL displays Ref BLGA, indicating that you should use the ARVL minor command of the BLGx majors for these types of transactions.

For transactions that can execute *either* in Fast Path (IFP) *or* in message processing regions (depending upon the fast path input routing exit at your installation) this value (for TRXx majors) shows executions in MPP regions only. The ARVL minor command of the BLGx majors displays values for executions in IFP regions only.

ATIL
Displays the average length of the input messages associated with this transaction.

Type:
Minor of transaction majors

The average length displays in bytes.

ATIL does not apply to transactions which execute in Fast Path (IFP) regions only and displays the value, FP XCL (Fast Path exclusive) instead.

ATTR
Displays three two-character fields that describe the attributes of this transaction.

Type:
Minor of transaction majors

Format:
ATTR aa bb cc

aa
Indicates either:

CN - the transaction runs in conversational mode FP - Fast Path potential transactions XF - Fast Path exclusive transactions

bb
If RS, the transaction runs in response mode

cc
If MS, this transaction is a multi-segment transaction

For example, the display **CN... MS** describes a transaction, conversational and multi-segment, but not response mode.

BALG
Displays the number of the associated Fast Path application balancing group.

Type:
Minor of transaction majors

This command is for Fast Path exclusive transactions only. For other than Fast Path exclusive transactions, BALG shows **not FP X**.

CLAS
Displays the IMS scheduling class for this transaction.

Type:
Minor of transaction majors

The scheduling class is a decimal number in the range of 1-255.

CLAS does not apply to transactions that execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

For all transactions defined in a multiple systems coupling (MSC) environment, CLAS displays Remote.

CPRI
Displays the current IMS dispatching priority for this transaction.

Type:
Minor of transaction majors

The dispatching priority is a decimal number in the range of 0-14.

CPRI does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

ENQL
Displays the enqueued limit count for this transaction.

Type:
Minor of transaction majors

ENQL does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

IQLN
Displays the number of messages for this transaction currently waiting on the input queue.

Type:
Minor of transaction majors

Finding a long queue here can mean a user issued a /STOP command for the transaction or there is no message region available to process it.

For transactions that *must* execute in a Fast Path (IFP) region, IQLN displays Ref BLGA to indicate that you should use the IQLN minor command of the BLGx majors for this type of transaction.

For transactions that can execute *either* in Fast Path (IFP) *or* message processing regions (depending upon the Fast Path input routing exit at your installation) this value (for TRXx majors) shows executions in MPP regions only. The IQLN minor command of the BLGx majors displays values for executions in IFP regions.

LPRI
Displays the limit scheduling priority for this transaction.

Type:
Minor of transaction majors

This number is a decimal number from 0-4.

LPRI does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

MXRG
Displays the maximum region count for this transaction.

Type:
Minor of transaction majors

This number is a decimal number from 0-255.

MXRG does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

NPRI
Displays the normal scheduling priority for this transaction.

Type:
Minor of transaction majors

This number is a decimal from 0-14.

NPRI does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive).

PGMN
Displays the program name associated with this transaction.

Type:
Minor of transaction majors

PLIM
Displays the message processing limit count for this transaction.

Type:
Minor of transaction majors

PLIM does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive) instead.

PRCS
Displays the number of transactions of this type which were processed.

Type:
Minor of transaction majors

For transactions that must execute in a Fast Path (IFP) region, PRCS displays Ref BLGA to indicate that you should use the MCNT minor command of the BLGc majors for this type of transaction.

For transactions that can execute either in Fast Path (IFP) or message processing regions (depending upon the Fast Path input routing exit at your installation) this value (for TRXc majors) shows executions in MPP regions only. The MCNT minor command of the BLGc majors displays values for executions in IFP regions.

PRLM
Displays the parallel limit scheduling value for the transaction.

Type:
Minor of transaction majors

PRLM does not apply to transactions which execute in Fast Path (IFP) regions only, and displays the value FP EXCL (Fast Path exclusive) instead.

PTYP
Displays the program (PSB) type associated with this transaction.

Type:
Minor of transaction majors

The type is batch or online.

For Fast Path transactions, PTYP displays FP Batch or FPOnline.

RCTE
Displays the routing code number associated with a Fast Path application.

Type:
Minor of transaction majors

For Fast Path exclusive transactions.

For other than Fast Path exclusive transactions, RCTE shows **not FP X.**

SCHC
Displays the number of regions in which this transaction is currently scheduled.

Type:
Minor of transaction majors

Unless the transaction's PSB is simultaneously scheduled in two or more regions, this number cannot be greater than one.

For transactions that *must* execute in a Fast Path (IFP) region, SCHC displays Ref BLGA to indicate that you should use the SPST minor command of the BLGx majors for this type of transaction.

SERL
Displays the SYSGEN SERIAL option for a transaction.

Type:
Minor of transaction majors

The option is either YES or NO.

SSPC
Displays the counter associated with the transaction's suspend queue.

Type:
Minor of transaction majors

STAT
Displays the status of the transaction.

Type:
Minor of transaction majors

The status of the selected transaction is one of the following states:

Active
Currently scheduled.

Idle
Schedulable, but not currently active.

Locked
A /LOCK TRAN command is issued against the transaction.

Norgns
No region is assigned for this transaction.

Pstopped
A /PSTOP TRAN command is issued against the transaction.

Purged
A /PURGE TRAN command is issued against the transaction.

Queued
Input messages queued but is not scheduled.

Queuing
Fast Path transactions are queued because all Fast Path regions are in use.

Stopped
A /STOP TRAN command is issued against the transaction.

Suspend
In the suspend queue.

Ustopped
No longer available for processing.

TRST
Displays the current trace status of the transaction.

Type:
Minor of transaction majors

The current trace status of the selected transaction is either On or Off.

IMS conversation commands

You can issue a number of OMEGAMON for IMS commands to select and display information about IMS conversations.

IMS conversation major commands

Use the IMS conversation major commands to select IMS conversations.

CNVA
Selects all conversations

Type:
Major

CNVH
Selects all held conversations.

Type:
Major

CNVI
Selects all idle conversations.

Type:
Major

CNVL
Selects all specified conversations.

Type:
Major

CNVR
Selects all running/active conversations.

Type:
Major

IMS conversation minor commands

You can use the IMS conversation minor commands to display information about selected IMS conversations.

DTME
Displays date/time stamp of the conversation.

Type:
Minor of IMS conversation majors

The format of the stamp is *YDDDHHMM*, where

Y
is the last digit of the current year

DDD
is the julian date

HH
is the hour

MM
is the minute

If the conversation is not active, the DTME command displays 00000000.

SLEN
Displays length of the SPA this conversation uses.

Type:
Minor of IMS conversation majors

SLEN displays the size of the SPA whether it is in-core or DASD.

If the conversation is not active, SLEN displays 0.

STAT
Displays status of this conversation.

Type:
Minor of IMS conversation majors

The status of the conversation is active, hold, or idle.

STYP
Displays type of SPA this conversation uses.

Type:
Minor of IMS conversation majors

The SPA type is one of the following types:

COR
in core

DSK
on disk

and one of the following types:

VARL
variable length

FIXL
fixed length

TERM
Displays name of the LTERM that is running this conversation.

Type:
Minor of IMS conversation majors

For active conversations only.

If the conversation is not active, the name is blank.

TRAN
Displays name of the transaction that starts this conversation.

Type:
Minor of IMS conversation majors

For running conversations only.

If the conversation is not active, TRAN displays a . (period).

Figure 133 on page 217 shows the results of the conversation minor commands when used with the CNVA major command.

```
CNVA       011      012      013      014      015      016      017      018  +
slen        88
stat      idle     idle     idle     idle     idle     idle     idle     idle
styp  COR VARL COR VARL COR VARL COR VARL COR VARL COR VARL COR VARL COR VARL
term      L542
tran  CONVTRAN
dtme  51601654 0000 000 0000 000 0000 000 0000 000 0000 000 0000 000 0000 000
```

Figure 133. IMS conversation commands

Pool commands

You can issue a number of OMEGAMON for IMS commands to display the characteristics and usage of various IMS pools.

Communications external subsystem pool commands

Use the CESS command to select the communications external subsystem (CESS) pool.

CESS
Displays statistics about the use of the CESS pool.

Type:
Major

The CESS command displays information about the communications external subsystem (CESS) pool. It displays the total size of the pool in bytes, the free space available within the pool in bytes, the current usage in percent, the usage high water mark (the largest amount that is used since IMS startup), the number of free blocks and the size of the largest free block. (These last two items indicate the extent to which IMS can fragment the pool.) The following figure provides a sample CESS display.

```
 CESS  Total Size in bytes =         0      Free space in bytes =          0
+      Current storage used =        0      Utilization =               .00%
+      Number of Free buffers =      0      Largest free buffer =          0
+      Number of buffer sizes =      0      Overflow size in bytes =       0
+      Upper Expansion Limit =   2147M      Usage Highwater mark =     13424
 aloc  Used Space  -  None
 extd  Extendable storage pool, storage and iwait analysis
+      Selective dispatching dormant        Extension is in CSA
+      #Allocated (Current)=                #Iwaits (Current)=
+      #Allocated (Total)=                  #Iwaits (Total)=
+      Current extension size=              Current iwait area size=
+      Max extension size=                  Max total area/iwaits=
+      Largest Allocated=                   Max single req/iwaits=
 free  Free Space                                               0089F96C
+ 000  00001000 00000000 00A24000           *.........s .    *
```

Figure 134. CESS display

The following minor commands display information about the communications external subsystem pool.

ALOC
Displays a memory dump of all the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of CESS

BUFS
Displays allocation and usage statistics for buffers in the CESS pool.

Type:
Minor of CESS

See a sample panel with the BUFS command, Figure 158 on page 246.

EXTD
Displays extended storage pool information.

Type:
Minor of CESS

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of CESS

Communications I/O pool commands

Use the `CIOP` command to display communications I/O pool (CIOP) information.

CIOP
Displays statistics about storage use within the communications I/O pool.

Type:
Major

Figure 135 on page 219 shows a typical CIOP major command and its minor commands.

```
 CIOP
 size  Statistics
+      Size Total   Alloc  Active   Free   High  RecAny LineBuf  PoolAv
+            4096     184   -na-    3912   3800   33280     129       0
+      Percents       4.4   -na-    95.5   92.7    12.5     3.1       0
+      Blocks
+        Numbers        2   -na-       1   -na-       8       1       0
+        Sml Size      32   -na-    3912   -na-    4160     129       0
+        Big Size     152   -na-    3912   -na-    4160     129       0
+        Avg Size      92   -na-    3912   -na-    4160     129       3
+      Node In      0  Node Out      0  Line In      1  Line Out      0
 extd  Extendable storage pool, storage and iwait analysis
+      Selective dispatching dormant        Extension is not in CSA        0
+      #Allocated (Current)=         8      #Iwaits (Current)=             0
+      #Allocated (Total)=           6      #Iwaits (Total)=               0
+      Current extension size=   33280      Current iwait area size=       0
+      Max extension size=       47604      Max total area/iwaits=         0
+      Largest Allocated=         7162      Max single req/iwaits=         0
```

Figure 135. CIOP command display

ALOC

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of CIOP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.

The following figure shows CIOP command output and displays free and allocated pool space.

```
 CIOP
 free  Free Space                                                  ADDR=91EAE0
+ 000  0000ED90 00000000 00164270          *............    *
 aloc  Used Space                                                  ADDR=91EABC
+ 000  00000020 0091EAA4 0015C000          *.....j.u..{.    *      ADDR=91EAA4
+ 000  00001038 0091EB04 0015C020          *.....j....{.    *      ADDR=91EB04
+ 000  00001038 0091EB10 0015D058          *.....j....}.    *      ADDR=91EB10
+ 000  00001038 0091EB1C 0015E090          *.....j....\.    *      ADDR=91EB1C
+ 000  00001038 0091EB28 0015F0C8          *.....j....0H    *      ADDR=91EB28
+ 000  00001038 0091EB34 00160100          *.....j......    *      ADDR=91EB34
+ 000  00001038 0091EB40 00161138          *.....j. ....    *      ADDR=91EB40
+ 000  00001038 0091EB4C 00162170          *.....j.<....    *      ADDR=91EB4C
+ 000  00001038 0091EB58 001631A8          *.....j.....y    *      ADDR=91EB58
+ 000  00000090 00000000 001641E0          *...........\    *
```

Figure 136. CIOP pool space

BUFS

Displays allocation and usage statistics for buffers in the CIOP pool.

Type:
Minor of CIOP

See a sample panel with the BUFS command, Figure 158 on page 246.

EXTD

Displays an analysis of the use of expandable storage pools.

Type:
Minor of CIOP

These pools expand and contract beyond storage limits which IMS startup execution parameters define.

EXTD analyzes the CESS, CIOP, and SPAP pools.

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of CIOP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

SIZE
Displays the communications I/O pool allocation and usage statistics.

Type:
Minor of CIOP

The following fields display on the screen when you enter CIOP with the SIZE minor command:

Total
The total amount of space available for the CIOP pool.

Alloc
The total amount of space that is allocated for CIOP buffer use Alloc does not include RECANY. See the RECA command for RECANY information.

Active
The total amount of space that is currently used in the CIOP pool.

Free
The total amount of space available for use.

High
The highest amount of used space.

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. BTAM buffers can be used for non-VTAM MSC links, non-VTAM MTO console, and non-VTAM secondary master console.

PoolAv
The amount of space available after the allocated space.

Scratch pad area pool command

Issue the SPAP command to display information about the Scratch Pad Area Pool (SPAP).

SPAP
Displays statistics about the utilizat of the Scratch Pad Area Pool.

Type:
Major

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of SPAP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

BUFS
Displays allocation and usage statistics for buffers in the SPAP pool.

Type:
Minor of SPAP

See a sample panel with the BUFS command, Figure 158 on page 246.

EXTD
Displays an analysis of the use of expandable storage pools.

Type:
Minor of SPAP

These pools expand and contract beyond storage limits which IMS startup execution parameters define.

EXTD analyzes the CESS, CIOP, and SPAP pools.

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of SPAP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

Receive Any (RECA) pool information

Issue the RECA command to display information about the Receive Any (RECA) pool.

RECA
Displays receive any pool information.

Type:
Immediate

```
RECA  Total size =      49152      Current utilization =     83.33%
+      Free space =      12259      HWM utilization =         91.67%
+      Buffer size =      4096      Number buffers in use =       10
+      Number of buffers =  12      HWM buffers in use =          11
```

Figure 137. Typical RECA command output display

High I/O pool (HIOP) information

Use the HIOP command to display information about the high I/O (HIOP) pool.

HIOP
Displays high I/O pool information.

Type:
Major

BUFS
Displays allocation and usage statistics for buffers in the HIOP pool.

Type:
Minor of HIOP

See a sample panel with the BUFS command, Figure 158 on page 246.

SIZE
Displays the automated operator interface pool allocation and usage sizes.

Type:
Minor of HIOP

The following figure shows typical HIOP major and SIZE minor output.

```
 HIOP
 size  Statistics
+                  Total   Alloc    Free    High
+      Size        81920   56033   25887   62587
+      Percent        na    68.4    31.6    76.4
+      Blocks
+       Numbers       na      17       1      na
+       Sml size      na    1024   25887      na
+       Big size      na    4096   25887      na
+       Avg size      na    3296   25887      na
```

Figure 138. Typical HIOP major and SIZE minor output

Field descriptions:

Total
Total size of the HIOP pool.

Alloc
Currently allocated pool storage.

Free
Currently available pool storage.

High
Highest amount of pool storage that is allocated as sampled by OMEGAMON for IMS.

Size
Amount of storage as represented by a specific column in the display.

Percent
Percent of storage as represented by a specific column in the display.

Blocks
The fields that follow display information about the pieces of contiguous storage in the pool.

Numbers
Number of blocks as represented by a specific column in the display.

Sml size
Smallest current block size for a specific column in the display.

Big size
Largest current block size for a specific column in the display.

Avg size
Average current block size for a specific column in the display.

Automated Operator Interface Pool

Use the AOIP command to display information about the Automated Operator Interface Pool (AOIP).

AOIP
Stores messages, commands, and responses instead of the message queues.

Type:
Major

BUFS
Displays allocation and usage statistics for buffers in the AOIP pool.

Type:
Minor of AOIP

SIZE
Displays the automated operator interface pool allocation and usage sizes.

Type:
Minor of AOIP

The following figure shows typical AOIP major command and SIZE minor command output.

```
 AOIP
 size  Statistics
+                  Total   Alloc    Free    High
+      Size        81920   56033   25887   62587
+      Percent        na    68.4    31.6    76.4
+      Blocks
+       Numbers       na      17       1      na
+       Sml size      na    1024   25887      na
+       Big size      na    4096   25887      na
+       Avg size      na    3296   25887      na
```

Figure 139. Typical AIOP major and SIZE minor output

Field descriptions:

Total
Total size of the AOIP pool.

Alloc
Currently allocated pool storage.

Free
Currently available pool storage.

High
Highest amount of pool storage that is allocated as sampled by OMEGAMON for IMS.

Size
Amount of storage as represented by a specific column in the display.

Percent
Percent of storage as represented by a specific column in the display.

Blocks
The fields that follow display information about the pieces of contiguous storage in the pool.

Numbers
Number of blocks as represented by a specific column in the display.

Sml size
Smallest current block size for a specific column in the display.

Big size
Largest current block size for a specific column in the display.

Avg size
Average current block size for a specific column in the display.

Main work area pool

Issue the WKAP command to display information about the IMS main work area pool.

WKAP
Displays size, free space, usage, and the usage high water mark of the general work pool.

Type:
Major

The following example shows the results of the WKAP command.

```
 WKAP  Total Size in bytes =     39056      Free space in bytes =        37152
       Utilization =             4.87%      Usage Highwater mark=         7112
```

Figure 140. Results of WKAP command

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of WKAP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP and QBUF, do not use the FAQEs to track memory.)

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of WKAP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP and QBUF, do not use the FAQEs to track memory.)

Database pool commands

Use the DBWP and DMPL commands to select and display information about the database pools.

DBWP
Displays statistics about the usage of the database work pool.

Type:
Major

Displays the total size of the pool in bytes, the free space available within the pool in bytes, the current usage in percent, the usage high water mark (the largest amount that is used since IMS startup), the number of free blocks and the size of the largest free block. (These last two items indicate the extent to which IMS can fragment the pool.) The following figure is a sample DBWP display.

```
 DBWP  Total Size in bytes =      8192      Free space in bytes =         8192
+      Utilization =              .00%      Usage Highwater mark =           0
+      Number of Free blocks =       1      Largest free block =          8192
```

Figure 141. Sample display

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of DBWP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of DBWP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

DMPL
Displays statistics about the database management block (DMB) pool.

Type:
Major

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of DMPL

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of DMPL

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

SIZE
Displays pool storage usage statistics for a specific storage pool.

Type:
Minor of DMPL

The following figure shows sample DMPL major and SIZE minor command output.

```
 DMPL
 size  Utilization, Current:    5.2%  Highest:       5.2%
+      Size Total   Alloc  Active     Free     Res Res-act G-total
+           16384     856            15528    1040           17424
+      Percents       5.2             94.7     5.9
+      Blocks
+        Numbers        1                1       3
+        Sml Size
+        Big Size     856            15528     712
+        Avg Size     856            15528
```

Figure 142. DMPL major and SIZE minor commands

The output includes the following information.

size
Storage information for DMB pool.

Current
% storage pool space currently allocated.

Highest
% storage pool space highest ever allocated.

Total
Total storage within DMB pool.

Alloc
Amount of storage that is allocated within pool.

Active
Amount of storage that is associated with active DMBs.

Free
Amount of deallocated storage within pool.

Res
Storage outside of pool that is associated with resident DMBs.

Res-act
Storage that is associated with resident active DMBs.

G-total
Total IMS storage for DMBs, sum of (Total + Res) values.

Percent
Percent of storage as represented by a specific column in the display.

Blocks
The fields that follow display information about the pieces of contiguous storage in the pool.

Numbers
Number of blocks as represented by a specific column in the display.

Sml size
Smallest current block size for a specific column in the display.

Big size
Largest current block size for a specific column in the display.

Avg size
Average current block size for a specific column in the display.

VSAM database buffer pool statistics

Use the `DBVS` command to display information about the VSAM buffer subpools.

The `DBVS` command displays the subpool number, the buffer size for the subpool, the number of buffers that are allocated, and indicates whether the buffers or IOBs are page-fixed. Buffer statistics similar to those statistics that the `/DISPLAY POOL DBAS` command displays also display, such as retrieves by key, records altered, and VSAM reads.

The `DBVS` command also lists the subpool ID and the databases that are assigned to the subpool, if it has specific databases that are assigned to it.

DBVS
Displays statistics that relate to individual VSAM buffer subpools.

Type:
Major

Format:
cDBVSnn

c
Leave *c* blank to display the statistics for VSAM buffer subpools or enter S to display summary information.

nn
VSAM subpool ID. If you do not specify an ID, DBVS displays data for all subpools.

The following example shows the summary that is displayed with SDBVS.

```
SDBVS   VSAM Subpool Summary                20 buffers totaling 63488 bytes
+        Retrieves by RBA =       13        Retrieves by Key =              15
+        Records altered =         0        Records created =                0
+        VSAM reads =           3231        VSAM writes =                    0
+        Found in pool =        2396        Sync points taken =              0
+        Buffers in error =        0        Maximum errors =                 0
+        Hit ratio =            74.1%       # VSAM strings =                11
+        VSAM strings active =     1        Max strings active =             1
```

Figure 143. Summary displayed with SDBVS

DBVS displays LSR pool information. The following figure shows an example of the displayed LSR pool information.

```
 DBVS03 VSAM Subpool #  3                   4 buffers of 2048 bytes each
+        Pool id: POL1                      Pool number: 0
+        Subpool type: Index                Subpool number within pool: 2
+        Buffers are not page-fixed         Blocks are not page-fixed
+        Retrieves by RBA =       10        Retrieves by Key =               0
+        Records altered =         0        Records created =                0
+        VSAM reads =            531        VSAM writes =                    0
+        Found in pool =         396        Sync points taken =              0
+        Buffers in error =        0        Maximum errors =                 0
+        Hit ratio =            74.5%
+        Buffers are backed by hiperspace
+        Data bases assigned to this subpool:
+        HSSP01FX(2) KD#IGW03(1) KD#MEM01(3) VDBSINDX(1)
```

Figure 144. Display VSAM buffer subpool statistics

The number in parentheses after the database name denotes the data set as specified in DFSVSMxx. For example, the number 2 in parentheses after HSSP01FX in Figure 144 on page 227 means that the second data set statement in the database is assigned to the subpool.

VSAM database subpool statistics

Use the VSUB command to display information about VSAM subpools.

VSUB
Displays VSAM subpool statistics.

Type:
Major

Format:
`VSUB n n n . . .`

The variable *n* is the number of each subpool that you want to analyze.

The minors of VSUB are described as follows.

BALT
Displays the number of logical records that are altered.

Type:
Minor of VSUB

BSIZ
Displays the size of the buffers in this subpool.

Type:
Minor of VSUB

CERR
Displays the number of error buffers currently in the subpool.

Type:
Minor of VSUB

FWRT
Displays the number of writes that are forced by VSAM.

Type:
Minor of VSUB

The number of non-user-initiated writes.

GETS
Displays the number of VSAM GET calls that are issued.

Type:
Minor of VSUB

ISES
Displays the number of logical inserts to ESDS.

Type:
Minor of VSUB

ISKS
Displays the number of logical inserts to KSDS.

Type:
Minor of VSUB

IWRT
Displays the number of user-initiated writes by VSAM.

Type:
Minor of VSUB

NBFR
Displays the number of buffers in this subpool.

Type:
Minor of VSUB

RKEY
Displays the number of VSAM retrieves by key.

Type:
Minor of VSUB

RRBA
Displays the number of VSAM retrieves by RBA.

Type:
Minor of VSUB

SCBF
Displays the number of schedule buffer calls issued.

Type:
Minor of VSUB

SFND
Displays the number of successful buffer finds.

Type:
Minor of VSUB

TERR
Displays the largest number of error buffers during this execution of IMS.

Type:
Minor of VSUB

TIO
Displays the total number of VSAM I/Os in this subpool.

Type:
Minor of VSUB

TWRT
Displays the total number of writes.

Type:
Minor of VSUB

VRDS
Displays the number of reads completed by VSAM.

Type:
Minor of VSUB

ISAM/OSAM buffer pool commands

Use the ISAM/OSAM buffer pool commands to display information about ISAM/OSAM buffer subpools.

The IMS systems programmer defines various numbers and sizes of subpools that are used for database I/O. Specifying the size and number of buffers in a subpool has a direct effect on the performance of database I/O. Use the ISAM/OSAM buffer pool commands to display database names, subpool IDs, and other statistics about the ISAM/OSAM buffer subpools.

SPAL
Displays database names and data set numbers of those databases that are assigned to specific subpools.

Type:
Immediate

Format:
SPAL cccc

cccc
Specifies a subpool ID. If you do not specify a subpool ID, SPAL displays all databases and data set numbers that are assigned to a buffer subpool in the DFSVSMnn member.

DBOS
Displays statistics about ISAM/OSAM buffer subpools.

Type:
Major

Format:
DBOSnn

nn
Specifies a subpool. If you omit the argument *nn*, data displays for all subpools and OMEGAMON for IMS produces a summary (consisting of totals for all subpools) at the end.

The DBOS command displays the subpool number, the buffer size for the subpool, the number of buffers that are allocated, and whether the buffers or buffer prefixes were page-fixed. Buffer statistics similar to those statistics from the `/DISPLAY POOL DBAS` command display, that is, locate calls, locates found in pool, and read requests.

DBOS also lists the subpool ID and databases that are assigned to the subpool, if it has specific databases that are assigned to it. The hit ratio field is a ratio of locates found in the pool to the locate calls.

The following example shows the results of the DBOS command.

```
DBOS02 ISAM/OSAM Subpool #  2              4 buffers of 2048 bytes each
+        Subpool id: AYCD
+        Buffers are not page-fixed         Prefix is not page-fixed
+        Locate Calls =        36033        Locates found in pool =
+        Read requests =        3835        Single block writes =
+        New Block requests=       0        Records altered =
+        Purge requests =        510        Blocks written by purge =
+        Total I/O errors =        0        Buffers locked by errors =
         Hit ratio =             5.0%
+    Data bases assigned to this subpool
+        AYCD01FX(1) DB$HFS01(1) DB$MDL03(2) SDBHINDX(4)
```

Figure 145. ISAM/OSAM buffer pool statistics

Note: The number in parentheses after the database name denotes the data set as specified in the DBDGEN. For example, the number 2 in parentheses after DB$MDL03 in Figure 145 on page 230 means that the second data set statement in the database is assigned to the subpool.

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of DBOS

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of DBOS

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

ISAM/OSAM subpool commands

Issue the OSUB command to display detailed statistics about the OSAM subpool.

OSUB
Displays detailed OSAM subpool statistics.

Type:
Major

Format:
`OSUB n n n . . .`

The variable *n* is the number of each subpool that you want to analyze.

The **OSUB** minor commands are as follows:

ABWP
Displays the average number of blocks that are written per purge operation.

Type:
Minor of DBOS

AWAT
Displays the average number of waits per subpool I/O request.

Type:
Minor of DBOS

A subpool I/O request is a read, write, or purge request.

BSZE
Displays the size of buffers in this subpool.

Type:
Minor of OSUB

EIO
Displays the number of I/O errors.

Type:
Minor of OSUB

LBFR
Displays the number of buffers locked because of I/O errors.

Type:
Minor of OSUB

LFND
Displays the number of locates found within this pool.

Type:
Minor of OSUB

NBUF
Displays the number of buffers in this pool.

Type:
Minor of OSUB

PSZE
Displays the amount of storage in use for prefixes and subpool control blocks.

Type:
Minor of OSUB

PURG
Displays the number of purge requests within the subpool.

Type:
Minor of OSUB

PWRT
Displays the number of purge writes.

Type:
Minor of OSUB

RIO
Displays the number of read I/O requests.

Type:
Minor of OSUB

SEAR1
Displays the number of buffers searched by locate.

Type:
Minor of OSUB

SWRT
Displays the number of single write requests.

Type:
Minor of OSUB

TLS
Displays number of buffers that are searched to find a buffer with a level *n*.

Type:
Minor of OSUB

Format:
TLSn

n is one of the following levels:

0
The buffer is available for use.

1
The buffer is empty.

2
A PST referenced this buffer.

3
A PST referenced this buffer and the buffer is busy (because the PST is doing I/O).

4
A PST referenced the buffer and altered its contents.

5
Something other than a PST referenced the buffer and altered its contents.

6
A PST is using the buffer but the PST does not alter the buffer's contents.

7
A PST is using the buffer and the PST altered the buffer's contents.

8
The buffer is busy reading.

TLV
Displays the number of buffers in use at level *n*.

Type:
Minor of OSUB

Format:
TLVn

n is one of the following levels:

0
The buffer is available for use.

1
The buffer is empty.

2
A PST referenced this buffer.

3
A PST referenced this buffer and the buffer is busy (because the PST is doing I/O).

4
A PST referenced the buffer and altered its contents.

5
Something other than a PST referenced the buffer and altered its contents.

6
A PST is using the buffer but the PST does not alter the buffer's contents.

7
A PST is using the buffer and the PST altered the buffer's contents.

8
The buffer is busy reading.

WLCT
Displays the number of locate calls that are waiting.

Type:
Minor of OSUB

WOWN
Displays the number of buffer steals or purge waits for ownership to be released.

Type:
Minor of OSUB

WSPL
Displays the number of steal waits when no buffers are available.

Type:
Minor of OSUB

WSTR
Displays the number of steal waits for busy reading.

Type:
Minor of OSUB

WSTW
Displays the number of steal waits for busy writing.

Type:
Minor of OSUB

WTOT
Displays the total number of waits for synchronization.

Type:
Minor of OSUB

Sequential buffering (SBUF) information

Use the SBUF command to display information about sequential buffering. Sequential buffering is a buffering technique that IMS can use to increase the speed of I/O processing for OSAM databases.

SBUF
Displays sequential buffering information.

Type:
Immediate

```
SBUF  Maximum storage = 76120      Current utilization = 28.04%
       Current storage = 21346      HWM utilization     = 37.76%
```

Figure 146. Sequential buffering (SBUF) information

Field descriptions:

Maximum storage
The maximum amount of storage that can be used for sequential buffering. SBUF displays **-no max-**if there is no maximum value that is provided in the IMS SBONLINE control card.

Current utilization
Percent of maximum storage in use.

Current Storage
Amount of storage in use for sequential buffering.

HWM utilization
Largest percent of maximum storage that is used for sequential buffering.

Fast path buffer pool commands

Use the fast path buffer pool commands to display information about the Expedited Message Handler Buffers (EMHB) pool and Fast Path Work Pool (FPWP).

EMHB
Displays statistics for the Expedited Message Handler Buffers (EMHB) pool.

This pool is in ECSA acquired by the IMS control region.

Type:
Major

BUFS
Displays allocation and usage statistics for buffers in the EMHB pool.

Type:
Minor of EMHB.

See a sample panel with the BUFS command, Figure 158 on page 246.

FPWP
Displays statistics for the Fast Path Work Pool (FPWP).

This pool is in IMS control region extended private.

Type:
Major

BUFS
Displays allocation and usage statistics for buffers in the FPWP pool.

Type:
Minor of FPWP

See a sample panel with the BUFS command, Figure 158 on page 246.

FPDB
Displays fast path buffer pool statistics as shown in the following figure.

Type:
Major

```
 FPDB  Fast Path Buffer Pool Statistics
 bsts
+      DBBF (Total FP Buffers Defined)  =        30
+      DBFX (FP Buffers for System Use) =        10
+      BSIZE (Size of each FP Buffer)   =      2048
+      Number of Regions Waiting for free Buffers = 0
 size
+                Total   Unfix     Pgfix    Pgfix    Pgfix    Pgfix     Pgfix
+                DBBF   New PSTs   SDEPS   PST Use   Avail    Int Q     OTHR
+
+      Buffers      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 147. FPDB major and BSTS and SIZE minor commands

BSTS
Displays fast path buffer pool (FPDB) statistics.

Type:
Minor of FPDB

The statistics are as follows:

DBBF
Total number of fast path buffers defined.

DBFX
Total number of fast path buffers defined as a cushion that allows for system and output thread usage.

BSIZE
VSAM control interval (CI) size of each fast path buffer. Possible CI sizes are 512, 1024, 2048 and 4096 bytes...up to 28K. Each buffer is the same size.

Also displayed are the number of regions that are waiting for a free fast path database buffer for the reading of a CI. This condition is commonly referred to as “hard luck wait”. Do not confuse this condition with a waiting for a CI currently in use by another region (or PST) or by an output thread.

SIZE
Displays the fast path buffer pool allocation and usage sizes.

Type:
Minor of FPDB

Sizes are by number of CIs, buffers, bytes, or percents as follows.

Total or DBBF
Total Buffers in the fast path buffer pool

Unfix new PSTs
Total number of fast path buffers defined that are not page-fixed for use by regions or by output threads. Unfix New PSTs is the number of buffers that are available for the normal buffer allocations (NBA) for more regions (IFP or MPP and BMP regions with NBA specified). If the number in the unfix new PSTs field is less than the normal buffer allocation that the dependent region JCL specifies, the additional region abends.

Pgfix SDEPS
Number of page fixed AREAs that are open and have a sequential dependent control interval (SDEP CI) defined. Each SDEP CI stores in a buffer. This buffer is acquired at AREA open and is written back to DASD when it becomes full (another buffer replaces it for the next SDEP CI) or when the AREA is closed.

Buffers are taken from the page fixed available allocation, but when buffers for the current SDEP CI are newly acquired, they cause fast path to recalculate the boundary between page fixed and unfixed buffers. If this boundary is adjusted, an unfixed buffer is given to the page fixed available allocation.

The opening of AREAs with SDEPs defined ultimately reduces the number of unfixed buffers available for the normal buffer allocation (NBA parameter on the dependent region JCL) for new PSTs. In determining the total number of fast path buffers (DBBF parameter on the control region JCL), consider the largest number of concurrent open AREAs with SDEPs defined.

Pgfix PST Use
Buffers that dependent regions are currently using for normal buffer allocation (NBA) and overflow buffer allocation (OBA).

Pgfix Avail
Buffers allocated but not currently in use for dependent region normal buffer allocation (NBA) and overflow buffer allocation (OBA). Pgfix Avail also includes fixed buffers that remain for system or output thread usage but are not in use for these purposes, that is, are not allocated for dependent region NBA or OBA.

Pgfix Int Q
Number of buffers that were previously used and released, and are now in the intermediate queue, where they await return to the page fixed available allocation.

The output thread that writes the update back to DASD releases these buffers to the intermediate queue. Unmodified buffers that the application process used are released to the intermediate queue during PST sync point.

Pgfix OTHR
Buffers that are output threads currently writing back to DASD.

EPCB pool information

Use the EPCB command to display information about the extended PCB pool (EPCB).

EPCB
Displays EPCB (extended PCB) pool information as shown in the following figure.

Type:
Major

```
 EPCB
 size  Statistics
+                 Total   Alloc    Free    High
+      Size       81920   56033   25887   62587
+      Percent       na    68.4    31.6    76.4
+      Blocks
+       Numbers      na      17       1      na
+       Sml size     na    1024   25887      na
+       Big size     na    4096   25887      na
+       Avg size     na    3296   25887      na
```

Figure 148. Typical EPCB major and SIZE minor output

SIZE
Displays the extended PCB pool allocation and usage sizes.

Type:
Minor of EPCB

Message format services (MFS) pool commands

Issue the Message format services (MFS) pool commands to display information about the fetch request elements (FRE) pool and message format pool (MFP).

FREP
Displays information about the current usage of the fetch request elements (FRE) pool.

Type:
Major

A FRE is in use if it is associated with a format block and if there is at least one immediate-fetch or pre-fetch user. There is no such thing as a FRE pool, but it is sometimes convenient to think of it that way. The user specifies the number of fixed FREs to allocate through the IMS FRE= startup parameter when IMS builds the message format block pool.

Because of the previous definition of FREs in use it is possible that the usage might be artificially low, and that you might require more FREs. To verify this, check the MFP display to see whether blocks are washing.

If all of these fixed FREs are ever in use at the same time the FREP command shows the pool as 100% used. (You can set the AFRE exception to warn when the usage exceeds some threshold.) If IMS require more FREs it must carve space for them out of the MFP buffer pool (these are dynamic FREs and are only kept around if they are in use). The FREP command also displays the number of dynamic FREs currently allocated; this is normally zero. Example:

```
FREP  Fixed FREs allocated =       20      Fixed FREs in use =             1
+      Utilization =            5.00%      Dynamic FREs =                  0
```

Figure 149. FREP command display

MFP
Displays message format pool usage statistics.

Type:
Major

ALOC

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of MFP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

BIOS

Displays pool I/O statistics, rates, and deltas for a specific pool.

Type:
Minor of MFP

FREE

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of MFP

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

REQU

Displays pool request statistics, rates, and deltas for a specific pool.

Type:
Minor of MFP

SIZE

Displays pool storage usage statistics for a specific storage pool as shown in the following figure.

Type:
Minor of MFP

Figure 150 on page 237 shows an example of MFP with SIZE, BIOS, and REQU command output.

```
 MFP   Statistics
================================================================================
 size   MFS Storage below 16m Line:   4464     MFS Directory Storage:   4912
+       MFS FRE Storage:     1448              MFBP Dynamic Area:   47696
+       Total - MFS Storage:   58520
+                   Total  Loading     I/f    Loaded  Free   Inact
+                   49152        0       0       0   43984    3520
+       Percents                .0      .0      .0    89.4     7.1
+       Blocks
+         Numbers                0       0       0    -na-      17
+         Sml size               0       0       0    -na-      64
+         Big size               0       0       0    -na-     696
+         Avg size               0       0       0    -na-     207
================================================================================
 bios          Total    I/f    P/f    Dir  Not-fnd Errors
+      Totals      39     18      0     21      3      0
+      Rates      .00    .00    .00    .00    .00    .00
+      Deltas       0      0      0      0      0      0
================================================================================
 requ           Total   I/f     P/f   No i/o  I/f wait  Washed  Ignored
+      Totals      125   101      24      103        0       0        4
+      Rates       .00   .00     .00      .00      .00     .00      .00
+      Deltas        0     0       0        0        0       0        0
```

Figure 150. Message format pool statistics

The output includes the following information.

size
Storage pool size information (minor command).

MFS Storage below 16m Line
The amount of storage that is below the 16m line.

MFS Directory Storage
Number of entries for directory I/O, each entry is 12 bytes.

MFS FRE Storage
Storage available for fetch request elements (FRE).

MFBP Dynamic area
Dynamic portion of MFP available for MFS definitions.

Total
Total size of MFP (sum of the previous values, plus 8).

Loading
Storage that is used for MFS pre-fetch while loading.

I/f
Storage that is used for MFS immediate fetch while loading.

Loaded
Storage that is used for pre-fetch, block is in memory.

Free
Available MFS pool storage.

Inact
MFS blocks in-core, no longer used.

bios
MFS I/O statistics.

I/f
Number of IMMEDIATE FETCH I/Os.

P/f
Number of PRE FETCH I/Os.

Dir
Number of Directory I/Os.

Not-fnd
Number of I/Os for entry not found.

Errors
Number of MFS I/O errors.

requ
MFS request analysis.

I/f
Number of IMMEDIATE FETCH requests.

P/f
Number of PRE FETCH requests.

No i/o
Number of requests that do not result in I/O.

I/f wait
Number of I/f waits for P/f to complete.

Washed
Number of blocks that are washed for an FRE.

Ignored
Number of P/f requests ignored.

```
MFP   STATISTICS
 ================================================================================
  size  MFS STORAGE BELOW 16M LINE:  2520     MFS DIRECTORY STORAGE:   5624
 +      MFS FRE STORAGE:   1824               MFBP DYNAMIC AREA:  18648
 +      TOTAL - MFS STORAGE:  28616
 +                   TOTAL  LOADING     I/F    LOADED  FREE   INACT
 +                   20480                            18424     224
 +      PERCENTS                                       89.9     1.0
 +      BLOCKS
 +        NUMBERS                                     -NA-        2
 +        SML SIZE                                    -NA-
 +        BIG SIZE                                    -NA-      160
 +        AVG SIZE                                    -NA-      112
 ================================================================================
  sb16  AREA DESCRIPTION            LOCATION       SIZE
 +      FIXED CONTROL AREA          00005360    1816 (000718)
 +      I/O STAGING BUFFERS         00005038     704 (0002C0) (1 BUFFERS)
 +       TOTAL - MFS STORAGE BELOW 16M LINE:    2520
 +      -------------------------------------------------------------------------
  sdir  AREA DESCRIPTION            LOCATION       SIZE
 +      PDS DIRECTORY INDEX         01D06F98     104 (000068) (8 ENTRIES)
 +      DYN. DIRECT. HASH TABLE     01D06A08      72 (000048)
 +      PRIMED DIRECTORY $$IMSDIR   01D06A50    1352 (000548) (74 ENTRIES)
 +      DYNAMIC DIRECTORY #0        01D05000    4096 (001000) (0 ENTRIES)
 +            TOTAL - MFS DIRECTORY STORAGE:    5624
 +      -------------------------------------------------------------------------
  sfre  AREA DESCRIPTION            LOCATION       SIZE
 +      FRE BLOCKS                  01D00000    1760 (0006E0) (40 ENTRIES)
 +      FRE HASH TABLE              01D006E0      64 (000040)
 +                  TOTAL - MFS FRE STORAGE:    1824
 +      -------------------------------------------------------------------------
  sdyn  AREA DESCRIPTION            LOCATION       SIZE
 +      MFBP DYNAMIC AREA           01D00720   18648 (0048D8)
 +      -------------------------------------------------------------------------
```

Figure 151. SB16, SDIR, SFRE, and SDYN minor commands

The SDIR command displays the following information:

- The PDS DIRECTORY INDEX field shows the location and size of the incore PDS directory index. This PDS directory index represents an index of all concatenated format libraries of the current format ddname (FORMATA or FORMATB). ENTRIES specifies the number of PDS members (format block members) represented by the PDS directory index.
- The DYN. DIRECT. HASH TABLE field shows the location and size of the incore dynamic directory hash table. The hash table is used to find entries in the dynamic directories.
- The PRIMED DIRECTORY $$IMSDIR field shows the location and size of the incore $$IMSDIR.
- The DYNAMIC DIRECTORY #0 field shows the location and size of incore dynamic directory number zero (#0). If an entry for a format is not found in $$IMSDIR, an entry is created in a dynamic directory when the format is first used. There can be up to 11 dynamic directories (0-A). ENTRIES indicates the number of format blocks that are represented by this dynamic directory.

PSB work pool commands

Use the PSB work pool commands to display information about the program specification block (PSB) pool.

PSBW

Displays statistics about the usage of the PSB work pool.

Type:

Major

Displays the total size of the pool in bytes, the free space available within the pool in bytes, the current usage in percent, the usage high water mark (the largest amount that is used since IMS startup), the number of free blocks and the size of the largest free block. (These last two items indicate the extent to which IMS can fragment the pool.)

```
 PSBW  Total Size in bytes =     12288      Free space in bytes =        10040
+      Utilization =            18.29%      Usage Highwater mark =        4496
+      Number of Free blocks =       1      Largest free block =         10040
```

Figure 152. IMS PSBW pool information

ALOC

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of PSBW

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

FREE

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of PSBW

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

PSPL

Displays program specification block pool statistics.

Type:
Major

The minor commands SIZE and DSIZ show in detail how IMS uses pool storage.

ALOC

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of PSPL

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

DSIZ

Displays pool storage usage statistics for a DL1-SAS PSB storage pool.

Type:
Minor of PSPL

FREE

Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of PSPL

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

REQU

Displays pool request statistics, rates, and deltas for a specific pool.

Type:

Minor of PSPL

The types of conflicts that are grouped under the REQU subparm OTHER are as follows:

- SMB locked or stopped
- PSB locked or stopped
- Database stopped
- PSB permanently bad

SIZE

Displays pool storage usage statistics for a specific storage pool.

Type:

Minor of PSPL

Figure 153 on page 241 shows an example of how to use the PSPL major command and its minors.

```
 PSPL
 size  Utilization, Current:  59.2%, Highest:  69.8%   (CSA storage)
+      Size Total   Alloc  Active    Free     Res Res-act G-total
+          65536   38832   12204   26704    1884          67420
+      Percents     59.2    18.6    40.7     2.7
+      Blocks
+        Numbers      14       2       4       1
+        Sml Size    336    3932      64    1876
+        Big Size   9520    8272   20928    1876
+        Avg Size   2773    6102    6676    1876
 dsiz  Utilization, Current:  94.6%, Highest:  99.7%   (DL1-SAS storage)
+      Size Total   Alloc  Active    Free     Res Res-act G-total
+         368640  349064  145584   19576   58348         426988
+      Percents     94.6    39.4     5.3    13.6
+      Blocks
+        Numbers      13       2       3       1
+        Sml Size   5896   40696    1248   58340
+        Big Size 104888  104888   14816   58340
+        Avg Size  26851   72792    6525   58340
```

Figure 153. DSIZ and SIZE minor commands of PSPL

The output includes the following information.

size

Minor command that display storage information for PSB pool (in CSA).

dsiz

Minor command that display storage information for PSB space (in DLISAS).

Current

% storage pool space currently allocated.

Highest

% storage pool space highest ever allocated.

Total

Total storage within PSB pool.

Alloc

Amount of storage that is allocated within pool.

Active

Amount of storage that is associated with active PSBs.

Free

Amount of deallocated storage within pool.

Res

Storage outside of pool that is associated with resident PSBs.

Res-act
Storage that is associated with resident active PSBs.

G-total
Total IMS storage for PSBs, sum of (Total + Res) values.

Percent
The percent of storage as represented by a specific column in the display.

Blocks
The fields that follow display information about the pieces of contiguous storage in the pool.

Numbers
The number of blocks as represented by a specific column in the display.

Sml size
The smallest current block size for a specific column in the display.

Big size
The largest current block size for a specific column in the display.

Avg size
The average current block size for a specific column in the display.

OMEGAMON for IMS provides analysis about the partitioning of the PSB pool, with PSBs partitioned in the DLISAS address space, and common storage (CSA). The SIZE minor command of the PSPL command contains expanded information about the CSA PSB storage areas, and DSIZ shows information about DLISAS storage.

Program isolation enqueue commands

Issue the program isolation enqueue commands to display information about the Program Isolation (PI) enqueue pool.

PIEP
Displays information about PI enqueue pool usage.

Type:
Major

The CORE parameter of the IMSCTF macro specifies the size of the PI enqueue pool at IMS gen time; since the pool can grow, the user specifies both an increment and a maximum size.

IMS starts out with one increment of space and then adds more increments as it requires until it reaches the maximum. The **PIEP** command displays the percentage of the theoretical maximum that is in use. The **PIEP** command displays usage relative to both current and maximum pool sizes. You can set the APIE exception to produce a warning when this usage exceeds some threshold.

The following example shows the result of the **PIEP** major command before initialization.

```
PIEP                       Maximum  Current    Used(bytes)  Unused(bytes)
 +      Size                   16K       0K           0              0
 +      Percent of maximum    100.00      .00         .00         100.00
 +      Percent of current      -NA-   100.00         .00            .00
 +      Pool increment =          2K   Avg. Length of Search =
 +      Maximum Search =           0   Request Rate/** Initialized **
```

Figure 154. PIEP major command before initialization

The following example shows the result of the PIEP major command after initialization.

```
PIEP                        Maximum  Current    Used(bytes)  Unused(bytes)
+      Size                     16K       2K          8          2040
+      Percent of maximum    100.00    12.50        .04         99.95
+      Percent of current      -NA-   100.00        .39         99.60
+      Pool increment =          2K   Avg. Length of Search =     .33
+      Maximum Search =           1   Request Rate/Sec      =     .03
+                                             Delta/Cycle   =       3
```

Figure 155. PIEP major command after initialization

After initialization, the Request Rate/Sec and Delta/Cycle fields might display blanks or zeros, depending on IMS activity.

Message queue buffer pool commands

Issue the QBUF command to display information about the message queue buffer pool.

QBUF
Displays information about the message queue buffer pool.

Type:
Major

This buffer pool contains a fixed number of buffers of a fixed size; all three message queue buffer data sets (short message, long message, and queue blocks) share these buffers.

ALOC
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for allocated space within each pool.

Type:
Minor of QBUF

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

BIOS
Displays pool I/O statistics, rates, and deltas for a specific pool.

Type:
Minor of QBUF

FREE
Displays a memory dump of all of the free-allocated queue elements (FAQEs) for free space within each pool.

Type:
Minor of QBUF

The memory dump is in hexadecimal.

This memory dump provides size and virtual addresses of each free and allocated area within the pool. (Some pools, such as MFP, QBUF, and WKAP do not use the FAQEs to track memory.)

REQU
Displays pool request statistics, rates, and deltas for a specific pool.

Type:
Minor of QBUF

SIZE
Displays pool storage usage statistics for a specific storage pool.

Type:
Minor of QBUF

As the following figure shows, the minor commands SIZE, REQU, and BIOS provide details about the current usage of pool storage, queue manager request activity, and associated I/O activity.

```
 QBUF  Statistics
 size               Total   S-Msg   L-Msg   Q-Blk Not-Act
+                   25920    2592            2592   20736
+      Percents              10.0            10.0    80.0
+      Blocks
+        Numbers        10      1               1       8
+      Size of ALL Qbuf blocks is:     2592
 requ          Requests  Found  Enq/deq   Enqueue   Dequeue   Iwaits
+      Totals      51       2       4         4                   1
+      Rates
+      Deltas
 bios           Total  Reads  Writes Forced Chk-pt
+      Totals       2      2
+      Rates
+      Deltas
```

Figure 156. QBUF major and SIZE, REQU, and BIOS minor commands

The output includes the following information.

size
: Storage pool size information.

S-Msg
: Blocks currently allocated to a short message block.

L-Msg
: Blocks currently allocated to a long message block.

Q-Blk
: Blocks currently allocated to queue blocks.

Not-Act
: Blocks that are not currently assigned to any of the previous.

requ
: Queue manager requests.

Found
: Number of internal requests that are satisfied without I/O.

Enq/deq
: Number of total enqueue and dequeue (messages that are processed.)

Enqueue
: Number of message enqueues.

Dequeue
: Number of message dequeues.

Iwaits
: Number of IWAITS (should always be close to zero).

bios
: Queue manager I/O activity.

Total
: Total number of queue manager I/Os.

Reads
: Number of reads since IMS restart.

Writes
: Number of writes.

Forced
: Number of forced writes.

Chk-pt
: Number of checkpoint writes.

Save area prefix (SAP) pool command

Issue the SAPP command to display information about the save area prefix (SAP) pool.

SAPP

Displays number of dynamic and privileged SAPs that are allocated and currently in use, and the usage that results.

Type:
Major

This buffer pool contains a fixed number of buffers.

The following example shows information about the save area prefix (SAP) sets.

```
 SAPP  Dynamic SAPs allocated       22      Dynamic SAPs in use =           0
+      Utilization =              .00%      Privileged SAPs =              10
```

Figure 157. SAP sets

A pre-assigned SAP is one that IMS permanently assigns to a certain function, and a privileged SAP is one that IMS can use only for privileged events.

LUMC

Displays statistics for the LU6.2 Manager Common buffer pool (LUMC).

This pool is in ECSA acquired by the IMS control region.

Type:
Major

BUFS

Displays allocation and usage statistics for buffers in the LUMC pool.

Type:
Minor of LUMC

See a sample panel with the BUFS command, Figure 158 on page 246.

LUMP

Displays statistics for the LU6.2 Manager Private buffer pool (LUMP).

This pool is in IMS control region extended private.

Type:
Major

BUFS

Displays allocation and usage statistics for buffers in the LUMP pool.

Type:
Minor of LUMP

Figure 158 on page 246 is an example of the LUMP command and the BUFS statistics for that pool.

```
________________ ZMENU    VTM     OI-II       /C I51A   01/02/97  14:17:10
> Help/News PF1     Exit PF3     Keys PF5     Command Mode PF12    Colors PF18
 LUMP  Total Size in bytes =     66224      Free space in bytes =         45576
+      Current Storage Used      20536      Utilization =                31.00%
+      Number of Free buffers       17      Largest free buffer =          4104
+      Number of buffer sizes        9      Overflow size in bytes =          0
+      Upper Expansion Limit =   2147M      Usage Highwater mark =       103336
 bufs  Fixed Pool Buffer Statistics                  Primary   Secondary  Aloc
+       Size       Total      Used     Free     High   Bufs/Blk   Bufs/Blk  Init
+        136          0         0        0        0         32         32    N
+        264          0         0        0        0         32         16    N
+        520          0         0        0        0         32         16    N
+       1032          0         0        0        0         32         16    N
+       2056         16         1       15        2         16          8    N
+       3072         12         1       11        1         12         12    N
+       4104          0         0        0        0          8          8    N
+      33032          0         0        0        0          4          2    N
+      Totals        28         2       26       na        168        110
```

Figure 158. LUMP pool statistics

IMS logging analysis

You can issue a number of OMEGAMON for IMS commands to display information about IMS logging analysis.

- "DASD logging analysis" on page 246
- "Online log data sets" on page 248
- "Write-ahead log data sets" on page 250

DASD logging analysis

Use the LSYS command to display detailed information about the DASD logging environment.

LSYS
Displays an analysis of the DASD logging feature.

Type:
Major

The LSYS command is a major command that selects information about the DASD logging feature. LSYS minor commands display the following information:

- Data about the logging environment
- Logging statistics
- OLDS and WADS data set data

The following minor commands display detailed information about the DASD logging environment.

LBUF
Displays OLDS buffer definition and usage statistics.

Type:
Minor of LSYS

LENV
Displays DASD logging environment data.

Type:
Minor of LSYS

LSTA
Displays DASD logging statistics.

Type:
Minor of LSYS

OLDS

Displays information about the active primary and secondary OLDS log data sets.

Type:
Minor of LSYS

The OLDS command displays information which concern the currently active OLDS data set. To display other OLDS data sets, enter the command OLDSnn where *nn* equals the numerical suffix of the data set you want to see.

If dual OLDS logging is in effect, the OLDS command displays information from both the primary and secondary data sets. To display only OLDS primary data set information, enter POLDS where P (primary data set) is in the label field (column 1). To display OLDS secondary data set information only, enter SOLDS where S (secondary data set) is in the label field (column 1).

WADS

Displays information about the active primary and secondary WADS log data sets.

Type:
Minor of LSYS

The `WADS` command is similar to the OLDS command, but it displays information about the current WADS data set. To display other WADS data sets, enter the command WADSn where *n* equals the numerical suffix (0–9) of the data set.

To display the primary WADS data set, enter PWADS where P (primary data set) is in the label field (column 1). To display the secondary WADS data set, enter SWADS where S (secondary data set) is in the label field (column 1).

The following figures show, in two parts, the LSYS command and its minor commands.

```
 LSYS  IMS/VS DASD Logging Environment and Statistics
 =======================================================================
 lenv  OLDS Logging = DUAL             Auto Archive Limit =    1
+      OLDS sets Defined =    3        OLDS sets Active   =    3
+      OLDS sets Stopped =    0        OLDS sets in ERROR =    0
+      -----------------------------------------------------------------
+      WADS Logging = DUAL             WADS Defined    =    2
+      WADS In Use  =    2             Spare WADS left =    0
 =======================================================================
 lsta                                  Totals     Rates/Sec.       Delta
+      Total Log Records                30927           .00            0
+      Total Log Blocks                   236           .00            0
+      -----------------------------------------------------------------
+      Write Ahead Requests              4065           .00            0
+      DC Waits for Write Ahead             0           .00            0
+      ------------------------------------------------------------------
+      Output Buffer Waits                 43           .00            0
+      # System Checkpoints                26           n/a            0
+      ------------------------------------------------------------------
+      EXCPVRs to the WADS               4190           .00            0
+      2K Blocks Written to WADS         4915           .00            0
+      ------------------------------------------------------------------
+      WRITEs to the OLDS                 235           .00            0
+      READs from OLDS                      0           .00            0
 ========================================================================
 lbuf  Log Buffer size        =   23296  Buffer size without prefix =  22528
+      Log Buffers defined    =       5  Total log buffer pool size = 120832
+      Log Buffers available  =       5  Buffers allowed for reads  =      2
+      Buffers used for writes =      0  Buffers used for reads     =      0
```

Figure 159. (Part 1 of 2). LSYS and its minor commands

```
 olds  DDNAME = DFSOLP00 - Primary      Status = Current - Open
+      DSNAME = IMS.V500.OLP00
+      Unit address = 153               Volume = OMON22
+      Logical record length = 22524    Blocksize = 22528
+      I/O Count =      235             I/O Rate =     .00 per second
+      Number of BLOCKs =     600       Current BLOCK    =     235
+      Starting BLOCK   =       1       Ending BLOCK     =     600
+      ------------------------------------------------------------------
+      DDNAME = DFSOLS00 - Secondary    Status = Current - Open
+      DSNAME = IMS.V500.OLS00
+      Unit address = 153               Volume = IMS002
+      Logical record length = 22524    Blocksize = 22528
+      I/O Count =      235             I/O Rate =     .00 per second
+      Number of BLOCKs =     600       Current BLOCK    =     235
+      Starting BLOCK   =       1       Ending BLOCK     =     600
==========================================================================
 LSYS  IMS/VS DASD Logging Environment and Statistics
 =========================================================================
 wads  DDNAME = DFSWADS0 - Primary      Status = Current - Open
+      DSNAME = IMS.V500.WADS00
+      Unit address = 755               Volume = IMS100
+      Logical record length =  2080    Blocksize =  2080
+      I/O Count =     4190             I/O Rate =     .00 per second
+      ------------------------------------------------------------------
+      DDNAME = DFSWADS1 - Secondary    Status = Current - Open
+      DSNAME = IMS.V500.WADS01
+      Unit address = 756               Volume = IMS200
+      Logical record length =  2080    Blocksize =  2080
+      I/O Count =     4190             I/O Rate =     .00 per second
```

Figure 160. (part 2 of 2). LSYS and its minor commands

Online log data sets

Issue the ODDS command to select and display the online log data sets in the order of use. The OLDS data set that is active displays first.

ODDS
Displays the OLDS data sets.

Type:
Major

The following minor commands display more information about the online log data sets.

OBLB
Displays last block number in OLDS (logging end of file).

Type:
Minor of ODDS

OBLK
Displays OLDS capacity in blocks.

Type:
Minor of ODDS

OBLW
Displays latest block number that is written to the OLDS.

Type:
Minor of ODDS

OBLZ
Displays OLDS block size.

Type:
Minor of ODDS

OBST
Displays first block number in the OLDS.

Type:
Minor of ODDS

OCIO
Displays the current I/O count of the OLDS.

Type:
Minor of ODDS

OLRL
Displays OLDS logical record length.

Type:
Minor of ODDS

OPDN
Displays the primary OLDS ddname.

Type:
Minor of ODDS

OPST
Displays the status of the primary OLDS.

Type:
Minor of ODDS

OPUA
Displays the unit address of the primary OLDS.

Type:
Minor of ODDS

OPVL
Displays the volume of the primary OLDS.

Type:
Minor of ODDS

ORDR
Displays the OLDS sequence of use.

Type:
Minor of ODDS

OSDN
Displays the secondary OLDS ddname.

Type:
Minor of ODDS

OSST
Displays the status of the secondary OLDS.

Type:
Minor of ODDS

OSUA
Displays the unit address of the secondary OLDS.

Type:
Minor of ODDS

OSVL
Displays the volume of the secondary OLDS.

Type:
Minor of ODDS

Write-ahead log data sets

Issue the `WDDS` major command to select and display the write-ahead data sets (WADS) in the order of use.

WDDS
Displays the WADS data sets in their order of use.

Type:
Major

The following minor commands display more information about the write-ahead log data sets.

WBLK
Displays the WADS block size.

Type:
Minor of WDDS

WCIO
Displays the WADS I/O count.

Type:
Minor of WDDS

WLRL
Displays the logical record length of the WADS data set.

Type:
Minor of WDDS

WSTA
Displays the status of the WADS.

Type:
Minor of WDDS

The status is either open or closed.

WUCB
Displays the unit address of the volume that contains the WADS.

Type:
Minor of WDDS

WUSE
Displays use of the WADS.

The use of the WADS is primary, secondary, or spare

Type:
Minor of WDDS

WVOL
Displays the volume on which the WADS is stored.

Type:
Minor of WDDS

Fast Path resource commands

You can use OMEGAMON for IMS major, minor, and immediate commands to analyze Fast Path resources.

- "Summary of Fast Path information" on page 251
- "Balancing groups" on page 251
- "Data entry database areas" on page 255
- "Main storage database commands" on page 258

- “Fast Path output thread command” on page 260
- “Route code commands” on page 260

Summary of Fast Path information

Issue the FYSY immediate command to display the quantity of each type of Fast Path (IFP) region that is currently active.

FSYS

Displays summary information about Fast Path in your system.

Type:

Immediate

This immediate command displays the number of each type of Fast Path (IFP) region currently active.

Field descriptions:

IFPs

Fast Path regions

MD

Message-driven

UT

Fast Path utility

```
FSYS  IFPs (MD) active = 0              IFPs (UT) active = 0
 +      BALG Msgs processed = 0           BALG Msgs queued = 0
 +      BALG Msg Enq Rate = .00/sec       BALG Msg Deq Rate = .00/sec
```

Figure 161. FSYS command display

This command also displays the total number of Fast Path messages based on all balancing groups (BALGs) queued and processed. The number of Fast Path messages queued equals the number processed plus the number of messages waiting to process.

The Fast Path message counts and rates are based on *current* Fast Path balancing group counters, and as such represent *current values* only. These values can change because IMS clears Fast Path balancing group messages processed counters to zero when the last Fast Path region processing the application is stopped.

Balancing groups

You can use the balancing groups commands to select Fast Path balancing groups.

BLGA

Selects all balancing groups.

Type:

Major

BLGI

Selects all idle balancing groups.

Type:

Major

BLGO

Selects all operational balancing groups.

Type:

Major

Operational balancing groups are those groups that are active.

BLGQ
Selects all queuing balancing groups.

Type:
Major

Queuing balancing groups are those groups that have a queue of expedited message handler buffers (EMHBs).

BLGU
Selects all unused balancing groups.

Type:
Major

APST
Displays the number of active PSTs for this balancing group.

Type:
Minor of balancing group majors

ARVL
Displays the number of Fast Path message arrivals for the application.

Type:
Minor of balancing group majors.

Use the .R argument to display the rate in seconds of Fast Path message arrivals.

BFWT
Displays the number of DEDB buffer waits for all of the Fast Path PSTs that are processing messages for the balancing group.

Type:
Minor of balancing group majors.

This number shows messages currently processing. It does not include data from messages that completed. A PST's buffer wait counter is reset at synchronization point.

Buffer waits lengthen message processing elapsed time. A non-zero buffer wait counter is a potential explanation for a non-zero input queue count (the IQLN minor command for this major displays this).

To determine the buffer wait count for individual dependent regions, refer to the NUOW minor command for the RGNc major commands.

BSTA
Displays the status of the balancing group.

Type:
Minor of balancing group majors

The status of the BALG is one of the following states:

Unused
No active Fast Path (IFP) regions for the application this balancing group anchored.

Idle
All of the regions, which process the application anchored by this balancing group, are waiting for input messages.

Active
One or more of the Fast Path (IFP) regions, which process the application anchored by this balancing group, are currently processing an input message.

Queuing
All of the regions, which process the application anchored by this balancing group, are currently processing input messages and additional input messages are beginning to queue.

DEDC
Displays the number of DEDB application calls for all of the Fast Path PSTs that are processing messages for the balancing group.

Type:
Minor of balancing group majors

The number of application DEDB calls that this command shows are for messages currently in process. This number does not include data from messages that completed. A PST's DEDB application call counter resets at sync point.

DEDC (the number of DEDB calls issued by the application) partially indicates the amount of work an application performs. A relatively large number of DEDB application calls might explain an input queue on the balancing group (see the IQLN minor command of the balancing group majors for more details).

An application DEDB call might require the reading of many or no CIs from DASD to supply the information it requires. One application DEDB call does not necessarily equate to one DEDB CI read. Refer to the DEDR minor command for this major for more information about determining the number of DEDB CIs read.

To determine DEDB call count for individual dependent regions, refer to the DEDC minor command for the RGNc major commands.

DEDR
Displays the number of DEDB reads (from DASD) for all of the Fast Path PSTs that are processing messages for the balancing group.

Type:
Minor of balancing group majors

DEDR (the number of DEDB CI READs an application requires) partially indicates the amount of resources an application requires to process a message. A relatively large number of DEDB READs might explain an input queue on the balancing group (see the IQLN minor command of the balancing group majors for more details).

An application DEDB call might require the reading of none or many CIs from DASD to supply the information it requires. One application DEDB call does not necessarily equate to one DEDB CI READ. A DEDB READ count greater than DEDB call count, indicates that on the average, the application needs to read multiple CIs to satisfy one DEDB call. A DEDB READ count less than the DEDB call count indicates that on the average, the application is able to satisfy multiple calls from the same CI.

DEDR.A displays the average number of CI reads per DEDB call.

To determine DEDB READ count for individual dependent regions, refer to the DEDR minor command for the RGNc major commands.

EMHB
Displays the number of associated expedited message handling buffers.

Type:
Minor of balancing group majors

IQLN
Displays the current number of messages queued on the balancing group.

Type:
Minor of balancing group majors

MCNT
Displays the number of messages this balancing group processed since it became operational.

Type:
Minor of balancing group majors

MSDC
Displays the number of MSDB application calls for all of the Fast Path PSTs that are processing messages for the balancing group.

Type:
Minor of balancing group majors

The number of application MSDB calls that this command shows are for messages currently in process. This number does not include data from messages that completed. The MSDB application call counter for a PST resets at synchronization point.

NUOW
Displays the number of CI contentions for all of the PSTs that are processing messages for the balancing group.

Type:
Minor of balancing group majors

The number of CI contentions is the number of PSTs that must wait for a CI because another PST or an output thread already has it.

The number of CI resource contentions this command shows are for messages currently in process. This number does not include data from messages that completed. A PST's resource contention counter resets at synchronization point.

CI resource contentions degrade Fast Path performance. A non-zero CI resource contention count is a potential explanation for a non-zero input queue count (the IQLN minor command displays this count).

To determine the resource contention count for individual dependent regions, refer to the NUOW minor command for the RGNc major commands.

OVWL
Displays the number of PSTs that currently wait to obtain the overflow buffer allocation latch because they exceeded their normal buffer allocation.

Type:
Minor of balancing group majors

Waiting for the OBA latch degrades Fast Path performance. To avoid this, increase the normal buffer allocation (NBA) or change the application to stay within the NBA.

PSBN
Displays the PSB name of the application that this balancing group anchors.

Type:
Minor of balancing group majors

Unused means that there are no active Fast Path (IFP) regions that process input messages for the application that this balancing group anchors.

SPST
Displays the number of PSTs attached to this balancing group.

Type:
Minor of balancing group majors

These are the PSTs IMS can schedule to process messages that come to this balancing group.

WPST
Displays the number of PSTs that are queued to this balancing group.

Type:
Minor of balancing group majors

These are the PSTs waiting to process Fast Path input messages.

Data entry database areas

You can use the data entry database (DEDB) areas commands to select DEDB areas and display detailed information.

DEDA

Selects all DEDB areas.

Type:

Major

DEDH

Displays all DEDB areas that have an HSSP BMP active.

Type:

Major

DEDH displays all DEDB areas that are actively processing a High Speed Sequential Processing (HSSP) batch message processing (BMP) region.

DEDL

Displays all specified DEDB areas.

Type:

Major

DEDP/n

Selects all DEDB areas that match a pattern.

Type:

Major

The `.SPT` command sets the pattern. See the `.SPT` command for information about setting patterns. If DEDP does not find a /n value, it uses the first pattern supplied with the last `.SPT` command.

The following minor commands provide additional information about the data entry database area.

DEDU

Displays all DEDBs that are closed or stopped.

DEDUC displays DEDBs that are closed and stopped.

DEDUO displays DEDBs that are open and stopped.

DEDUS displays DEDBs that are stopped (open or closed).

Type:

Major

A DEDB is unusable if it is closed or stopped.

DSME

Displays Fastpath VSO DEDB area I/O rates for all DEDB areas allocated to dataspaces. By using the optional dataspace name, the list is limited to only those DEDB areas allocated to the specified dataspace.

Type:

Immediate

DSML

Displays all the Fastpath VSO dataspaces allocated to the IMS system as well as dataspace and DASD I/O statistics. Each dataspace is allocated as a result of defining the database as a VSO database in the Database Recovery Control (RECON) data set.

Type:

Immediate

DSPS

Displays summary information about the dataspace and the areas allocated to dataspaces. By using the optional dataspace name, the display is limited to a single dataspace.

Type:
Immediate

DSPX
Displays the number of dataspaces, number of Fastpath DEDB areas allocated to the dataspaces, the dataspaces with the highest and lowest dataspace I/O rate, and the areas with the highest and lowest I/O rate.

Type:
Immediate

AINF
Shows information about the selected DEDB area.

Type:
Minor of data entry database majors

The information displays in the following format:

CL .. ..
Area closed.

OP .. ..
Area opened.

ST .. ..
Area stopped and closed.

.. SD ..
Area has sequential dependent segments defined.

.. SF ..
Area has sequential dependent segments defined and during Fast Path synchronization point there is not enough room in the sequential dependent portion of the area to insert a segment.

.. .. ER
Area has an I/O error and requires recovery.

CIAB
Shows the number of anchor blocks (CIs) in each unit of work.

Type:
Minor of data entry database majors

CIDO
Shows the number of CIs in the dependent overflow portion of each unit of work.

Type:
Minor of data entry database majors

CIIF
Shows the number of free CIs in independent overflow.

Type:
Minor of data entry database majors

CIIO
Shows the number of CIs in independent overflow.

Type:
Minor of data entry database majors.

CIRA
Shows the number of CIs in the root addressable portion of the area.

Type:
Minor of data entry database majors

CISF
Shows the number of free CIs in sequential dependent area.

Type:
Minor of data entry database majors

CISO
Shows the number of CIs in sequential dependent area.

Type:
Minor of data entry database majors

CISZ
Shows the size of a CI in this area.

Type:
Minor of data entry database majors.

CIUW
Shows the total number of CIs in this UOW.

Type:
Minor of data entry database majors

CUOW
Shows the number of DEDB resource request conflicts.

Type:
Minor of data entry database majors

Each conflict comprises two or more requests.

FPUT
Shows the Fast Path utility currently active on the DEDB area.

Type:
Minor of data entry database majors

The possible values are as follows:

COMPARE
Fast Path online DEDB area data set compare (DBFUMMH0).

CREATE
Fast Path online DEDB area data set create (DBFUMRI0).

DELETE
Fast Path online sequential dependent delete (DBFUMDL0).

HSSP
High Speed Sequential Processing.

none
No Fast Path online utility active on the area.

REORG
Fast Path online reorganization (DBFUMDR0).

SCAN
Fast Path online sequential dependent scan (DBFUMSC0).

NAME
Shows the name of the DEDB of which this is an area.

Type:
Minor of data entry database majors

XCRB
Shows the active DEDB resource requests in this area.

Type:
Minor of data entry database majors

The XCRB control block represents the active requests. If the major command shows more than one area, XCRB shows only the first area that you list. These are the XCRB fields and what they mean:

Jobname
Name of region holding or waiting for resource. If name is OTHR, an output thread is holding the resource (control interval).

PSBName
Name of the program specification block

Trxname
Name of the transaction.

DBname
Name of the Fast Path DEDB.

AREAname
Name of this partition of the DEDB.

R.B.A.
Relative byte address - the address of the resource (control interval).

Status
Status of the resource request. Possibilities are:

EX/OWNER
exclusive/owner

EX/WAITING
exclusive/waiting - highlighted

NE/OWNER
non-exclusive/owner

NE/WAITING
non-exclusive/waiting - highlighted

P.I.
Is the resource control request known to IMS Program Isolation? YES or NO

Main storage database commands

Issue the main storage database commands to select main storage databases (MSDBs) and display detailed information.

MSDA
Selects all MSDBs.

Type:
Major

MSDL
Selects MSDBs by list.

Type:
Major

MSDP/n
Selects all MSDB names that match a pattern.

Type:
Major

The .SPT command sets the pattern. See the .SPT command for information about setting patterns. If MSDP does not find a /n value, it uses the first pattern supplied with the last .SPT command.

The following minor commands display additional information about main storage databases.

DLEN
Shows the virtual size of the entire MSDB (data).

Type:
Minor of main storage database majors

IPKD
Shows if the MSDB contains an invalid packed field.

Type:
Minor of main storage database majors

The status is either invalid or valid.

KYTP
Shows the key type of the MSDB.

Type:
Minor of main storage database majors

The key type is either segment or lterm.

OFLW
Shows if overflow occurred in a field.

Type:
Minor of main storage database majors

OMEGAMON for IMS displays either overflow or no ovflw.

OFWF
Shows a list of the fields that have overflowed.

Type:
Minor of main storage database majors

The fields display only for the first selected MSDB.

PGST
Shows the paging status of the MSDB.

Type:
Minor of main storage database majors

The paging status is either pageable or a fixed page.

R2V
Displays the MSDB real to virtual size percentage.

Type:
Minor of main storage database majors

For pageable MSDBs, R2V shows the percentage of the MSDB in real storage. For fixed MSDBs, R2V shows fixed.

SLEN
Displays the length of one MSDB segment.

Type:
Minor of main storage database majors

TYPE
Displays the organization type of the MSDB.

Type:
Minor of main storage database majors

The organization type is one of the following types:

non-rltd
not terminal related

fix rltd
fixed size, terminal related

dyn rltd
dynamic, terminal related

WKST
Displays the MSDB working set size.

Type:
Minor of main storage database majors

For pageable MSDBs, shows the number of bytes in real storage. For fixed MSDBs, shows fixed.

Note: For z/OS, the display includes expanded storage.

Fast Path output thread command

You can use the OTHR command to display information about Fast Path output threads including the number of defined output threads, active output threads, idle output threads, and buffers that are waiting for an output thread.

OTHR
Displays information about Fast Path output threads.

Type:
Immediate

The OTHR command is an immediate command that displays the following information:

```
 OTHR         Output Thread Information
+      Defined =    3   Active =    0   Idle =    3
+      Buffers waiting for an OTHR =    0
+      Total buffers queued on OTHRs =    0
```

Figure 162. OTHR command display

Route code commands

Issue the route code commands to select Fast Path routing codes and display detailed information.

RCDA
Selects all route codes.

Type:
Major

RCDL
Selects RCTEs by list.

Type:
Major

RCDO
Selects operational RCTEs.

Type:
Major

Operational route codes are those codes that are active.

RCDP/n
Selects all RCDBs that match a pattern.

Type:
Major

The .SPT command sets the pattern. See the .SPT command for information about setting patterns. If RCDP does not find a /n value, it uses the first pattern supplied with the last .SPT command.

RCDU
Selects unusable RCTEs.

Type:
Major

BALG
Displays the number of the associated BALG (if RCTE is active).

Type:
Minor of route code majors

PGNM
Displays the program name.

Type:
Minor of route code majors

STAT
Displays the status of the RCTE.

Type:
Minor of route code majors

The status is stopped, not schd (not scheduled), or active.

The following example show output from the RCDA command and its minors.

```
RCDA   FPSAMP1
BALG         1
PGNM  DBFSAMP3
STAT    active
```

Figure 163. RCDx command and its minors

Extended Recovery Facility (XRF) Support

Use the Extended Recovery Facility commands to display information about the XRF environment.

The XRF commands apply only to IMS systems that were installed with XRF. You can run the XRFS command on the active IMS system and the standby IMS system. The XRFT command only runs on the standby IMS system.

XRFS
Displays summary information about the XRF system.

Type:
Immediate

Figure 164 on page 261 shows the output from XRFS when it runs on the active IMS system. Figure 165 on page 261 shows the output from XRFS when it runs on the standby IMS system.

```
XRFS Active system
+     Class 1 Terminals                  = 350
+     Class 2 Terminals                  =  24
+     RSE name = IMSPROD          AVM is active
```

Figure 164. XRFS command executed on the active IMS system

```
XRFS Backup system
+     Terminals in backup mode           = 374
+     Class 1 Terminals                  = 350
+     Class 2 Terminals                  =  24
+     RSE name = IMSPROD          AVM is active
+     Active IMSID = IM22
+     Active CPUID = 0208570103084
```

Figure 165. XRFS command executed on the standby IMS system

The output from XRFS includes the following information.

System status
Indicates if this IMS system is currently an active or standby IMS system.

Eligible terminals
The number of terminals defined in the IMS gen as Class 1 or Class 2 terminals.

Backup terminals
The number of terminals logged on to the active IMS system with backup sessions on the standby IMS system. The number of terminals which must be switched from the active system to the standby system in the event of a takeover.

RSE name
The name of the recoverable service element (RSE). The RSE is the basic unit of control for XRF. The two IMS systems (active and standby) comprise the RSE.

AVM status
Indicates if the availability manager (AVM) is currently active or inactive. To insure integrity of data sets shared between the active and standby systems, XRF must stop all I/O activity when a takeover begins. This function is a function of the AVM. The AVM is a component of the z/OS system.

Active IMSID
The IMSID of the active IMS system. The XRFS command displays this field only when it is executed from the OMEGAMON for IMS monitoring the standby IMS system.

Active CPUID
The CPUID of the active IMS system. The XRFS command displays this field only when it is executed from the OMEGAMON for IMS monitoring the standby IMS system.

XRFT
Displays general information related to a pending XRF takeover.

Type:
Immediate

XRFT displays the following information:

Takeover status
Indicates if the IMS standby system is currently taking over the active IMS system.

Held PSTs
The number of dependent regions held on the standby system due to back-out.

Sessions not switched
The number of terminal sessions which still must be switched from the active to the standby system.

OMEGAMON for IMS can only execute this command when it is monitoring the standby IMS system.

XRSV
Displays the XRF surveillance status as shown in the following figure.

Type:
Immediate

```
 XRSV  Type   Status   Interval   Timeout
+      ----   ------   --------   -------
+      LINK   Active      2 sec     8 sec
+       LOG   Active      3 sec     9 sec
+       RDS    Inact     --na--    --na--
```

Figure 166. XRSV command output (RDS surveillance mode inactive)

```
 XRSV  Type   Status   Interval   Timeout
+      ----   ------   --------   -------
+      LINK   Not sel    --na--    --na--
+       LOG    Active     3 sec     9 sec
+       RDS    Active     4 sec     7 sec
```

Figure 167. XRSV command output (LINK surveillance mode not selected)

The headings in the display indicate:

Type
Type of monitoring

Link
Surveillance is done using an ISC link.

Log
Surveillance is done using the IMS system log.

RDS
Surveillance is done using the RDS data set.

Status
Status of XRF monitoring

Active
Surveillance is in progress.

Inact
Surveillance is not in progress.

Not_sel
Surveillance is not selected.

Interval
The monitoring interval (in seconds)

Timeout
The timeout value (in seconds)

Chapter 11. External subsystems information

You can issue OMEGAMON for IMS commands to monitor external subsystems (ESS) that are defined to IMS regions. Use the external subsystem major commands to select external subsystems and the minor commands to display detailed information about the selected external subsystems.

Major commands to monitor external subsystems

Use the following major commands to select external subsystems.

SUBA
Selects all external subsystems.

Type:
Major

SUBL
Selects all external subsystems listed.

Type:
Major

SUBP/n
Selects all external subsystems that match a pattern.

Type:
Major

Use the **.SPT** command to set the pattern.

Minor commands to monitor external subsystems

Use the external subsystems minor commands to display detailed information about selected external subsystems (ESS).

ECRC
Displays the command recognition character that is used to pass commands to the external subsystem.

Type:
Minor of external subsystem majors

PARM
Displays all dependent regions, which have the first displayed external subsystem that is defined and their associated parameters.

Type:
Minor of external subsystem majors

The PARM command displays the following information for the first subsystem that the major command lists:

- IMS region name
- Current external subsystem interface status for the region, including Init, Iden, Echo, Trm-Iden, Term, Resolve, Sign on, Cre-Thrd, Trm-Thrd, Sign off, Prepare, Abort, Not-Oper, SQL-Call, MQ-Call, Command, and Commit. The value IDLE is displayed when no activity is occurring.
- Language interface token
- Resource translation table name
- Interface control module name
- Error option specification

The current external subsystem interface status is one of the following states:

Init
External subsystem is initializing.

Iden
External subsystem initialized and is completing identify processing.

Signon
External subsystem is completing signon processing.

Trm-Iden
External subsystem is completing identify termination processing.

Signoff
DB2 external subsystem is completing signoff processing.

Note: This status is not applicable for MQ external subsystems.

Term
External subsystem is completing termination processing.

Echo
External subsystem is completing echo processing.

Resolve
External subsystem is completing indoubt resolution processing.

Cre-Thrd
External subsystem is creating a thread on behalf of an application.

SQL-Call
SQL call is in progress (DB2 external subsystems only).

MQ-Call
MQ call is in progress (MQ external subsystems only).

Command
SQL or MQ command is in progress.

Prepare
External subsystem is preparing to commit (phase 1 commit in progress).

Commit
External subsystem is completing commit processing (phase 2 commit in progress).

Abort
External subsystem is completing back out processing.

Trm-Thrd
External subsystem is ending a thread.

Not-Oper
External subsystem is not operational.

Idle
External subsystem that is connected, but inactive

STAT
Displays the status of the external subsystem.

Type:
Minor of external subsystem majors

The status is one of the following states:

Active
The external subsystem is doing work for an IMS dependent region.

Conn
The external subsystem is not doing work for any IMS dependent region.

Not-Con
The external subsystem is not connected to any IMS dependent region.

Not-Def
The external subsystem is not defined to any IMS dependent region.

Use the PARM minor command to determine which IMS dependent regions are using the external subsystem.

TYPE
Identifies the type of external subsystem: DB2, MQ, or Other.

Type:
Minor of external subsystem majors

Chapter 12. OMEGAMON for IMS authorized commands

The term *authorized commands* refers to certain sensitive or powerful commands that can alter data in storage and display data from restricted storage areas. Misuse of authorized commands can jeopardize your system and the integrity of your data. Use authorized commands with caution.

The OMEGAMON for IMS internal security facility provides security for all OMEGAMON for IMS commands. Each OMEGAMON for IMS command can have a security level of 0, 1, 2, or 3. Security level 3 provides the highest degree of protection; levels 2 and 1 provide successively reduced degrees of protection. A setting of 0 means that any user can access the command.

You can secure OMEGAMON for IMS commands internally or by setting up an interface with your external security system at your installation (for example RACF or ACF2®). The security facility is described in the *IBM OMEGAMON for IMS on z/OS: Planning and Configuration Guide.*

Some OMEGAMON for IMS commands ship with a security level of 3. These *authorized commands*, which are described in this section, can be accessed only by entering a password. All other commands ship with a security level of 0.

Issuing authorized commands

To access authorized commands, enter your password by issuing the `/PWD` INFO-line command. Before you use authorized commands, you should understand system access considerations, how to use the action character, how to access other address spaces, and how to specify addresses.

For information about how to use the `/PWD` command, see Chapter 2, "Operational commands for your OMEGAMON for IMS system," on page 11.

The execution of authorized commands might fail because the Resource Measurement Facility (RMF) Monitor I is not active. To see whether the RMF Monitor I is active, enter the `.RMF` immediate command.

.RMF
Displays Resource Measurement Facility information.

Type:
Immediate

The `.RMF` immediate command displays whether the Resource Measurement Facility (RMF) Monitor I is active and, if it is, the version number and release level.

Note: The amount of information that is displayed by the `.RMF` command varies according to the level of RMF that you are running.

```
:>.RMF    >> RMF Monitor I active <<
>        The RMF version code is 422F; RMF version 4.2.2
>        Current Interval Length:  14:56 MN   Cycle Time: 1000 ms
```

Figure 168. .RMF command display

Using the action character with authorized commands

For most OMEGAMON for IMS authorized commands, you must enter an action character in column 1 before OMEGAMON for IMS can run the command. The action character is a hyphen (-). Some commands require this character whenever you start them; others require the action character only on certain occasions.

If you omit the action character from a command that requires it, OMEGAMON for IMS displays a **No Act** message at the far right of the line. To remove the **No Act** message, type a hyphen in column 1 and press **Enter**. OMEGAMON for IMS then runs the command.

Table 11 on page 270 shows the authorized commands and whether they require the action character.

Table 11. Commands that require the action character

Command	Action Character Required
CONS	Never required.
CONU	Never required.
ACTN	Never required.
LINE	Never required.
MNT	Never required.
DYNA	Required to allocate data set.
DYNU	Required to allocate data set.
ICMD	Always required.
ICNS	Always required.
ILST	Required if storage is not in your key.
DYNU	Required if storage is not in your key.
ISCN	Required if storage is not in your key.
IZAP	Required if storage is not in your key.
MCHN	Never required.
MDEF	Never required.
MLOG	Always required.
MLST	Never required.
MSCN	Never required.
MZAP	Required if storage is not in your key.
OCMD	Always required.
OSPC	Never required.
PEEK	Required the first time you look at a job or need to collect new information. See “Collecting data about address spaces” on page 292 for complete information.
AMAP	Never required.
DATA	Never required.
DDNS	Never required.
JOBS	Never required.
MODS	Never required.
STEP	Never required.
SUBP	Never required.
TCBS	Never required.
SLST	Always required.
SSCN	Always required.
SCHN	Always required.

Table 11. Commands that require the action character (continued)	
Command	**Action Character Required**
SZAP	Always required.
XMCH	Always required.
XMLS	Always required.
XMSC	Always required.
XMZP	Always required.

The SRB/XMS facility

OMEGAMON for IMS has several authorized commands that must schedule an SRB to perform their function and some that must access other address spaces.

Examples of authorized commands that must schedule an SRB are **XMLS**, **XMSC**, and **XMZP**. Examples of authorized commands that must access other address spaces to complete their function are **PEEK**, **XMLS**, **XMSC**, and **XMZP**. Generally, OMEGAMON for IMS uses cross memory services (XMS) to access non-swappable address spaces and SRB routines to access all other address spaces.

Note: OMEGAMON for IMS completely recovers from all abnormal conditions; it should never cause a system dump or affect the IMS region in any way (other than to swap it into memory).

SRB timeouts

OMEGAMON for IMS waits only 10 seconds for a Service Request Block (SRB) to be dispatched. If the SRB does not dispatch in this period, OMEGAMON for IMS automatically purges the SRB and continues.

If OMEGAMON for IMS automatically purges the SRB and continues, OMEGAMON for IMS displays the following message.

```
       WARNING      SRB TIMED OUT AFTER 10.00 SECONDS (RC=8)
```

This message indicates that the IMS region cannot service OMEGAMON for IMS's SRB within the 10-second interval, so the SRB is purged. The problem might be only temporary, so you reenter the command that causes the condition.

SRB/XMS routine program checks

If the IMS region is damaged, some of its control blocks might be overlaid or contain invalid pointers or addresses. OMEGAMON for IMS monitors this condition by checking the ID field of each control block before use. If OMEGAMON for IMS finds a suspect control block, analysis is discontinued and an error message displays.

In some cases, this situation can cause the SRB/XMS routine to program check when it runs in the IMS region. OMEGAMON for IMS's SRB/XMS routines recover by using a functional recovery routine (FRR) when a program check occurs, and the issuing command displays a warning message as shown in the following figure:

```
WARNING      SRB ROUTINE TERMINATED ABNORMALLY - RC=S0C4
PSW AT TIME OF ERROR = xxxxxxxxxx xxxxxxxxxx
 + R0-8 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
 + R9-F xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
 + A0-8 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
 + A9-F xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

Figure 169. SRB/XMS routine program checks

This hexadecimal display shows the PSW and general registers at the time of the program check.

Address specification for storage commands

You can specify, modify, or predefine an address (*addr*) for commands that display or modify storage or data-only spaces (ESA). An address consists of an anchor, optional modifiers, and an optional predefined name.

Anchors

An anchor is the base address of an address specification. An anchor can be:

Absolute
The hexadecimal address.

Symbolic
Up to eight alphanumeric characters, including @, #, and $. See "Predefinitions" on page 272.

Modifiers

You can supply one or more modifiers to change the location that the anchor points to. A modifier can be:

Offset
A plus sign (+) or minus sign (-), followed by a hexadecimal number.

This modifier specifies a location at a known offset (positive or negative) from the anchor address.

Indirect
Use a question mark (?) as the symbol for 31-bit (XA or ESA) addressing.

This modifier indicates that the location pointed to is itself an address.

You can use these modifiers to create powerful and versatile address expressions. For example, the following address expression locates the TIOT of the z/OS task that is running:

```
10%%+4%+C%
```

This expression has the following components:

10%
Treats the data at locationX'10' as a 24-bit address. The address of the z/OS CVT.

%
The second % treats the data at the start of the CVT as a 24-bit address. The first word of the CVT contains the address of a doubleword (8 bytes). The doubleword contains:

- The address of the next TCB to dispatch (bytes 0 - 3)
- The address of the currently dispatched TCB (bytes 4 - 7)

+4%
Adds X04 to the address of the doubleword, and then treats the data at the displacement as a 24-bit address. This address is the address of the currently dispatched TCB.

+c%
Adds X 0C to the address of the TCB, and then treats the data at the displacement as a 24-bit address. This address is the address of the TIOT.

Note: This example does not work for commands that point to another address space (XMLS, XMZP, XMSC). Because this code is interpreted in the IMS region, there is no currently dispatched TCB to locate, and the pointer is zero.

Predefinitions

You can specify or reference an address by a name that consists of up to eight alphanumeric characters, including @, #, and $. The following command specifies address names:

MDEF
Defines names for addresses.

The MDEF command labels an address in storage with a name so that you do not have to repeatedly enter complex or frequently used storage addresses.

Type:
Immediate

Format:
MDEF cc addr,name:

cc
The type of operation.

CL
Clears the address name table, and resets it to initial status.

LP
Lists pre-defined address names.

LS
Lists the address names in the table.

DE
Deletes an address name.

bb
Assigns an address name.

addr
The address that you want to name. For more information, see "Address specification for storage commands" on page 272.

name
The address name. Up to eight alphanumeric characters, including @, #, and $.

Consider the following points:

- OMEGAMON for IMS saves the names that you define in a table, but does not save the table from session to session. Save address names that you want to use from session to session in a screen space.
- OMEGAMON for IMS provides a table of predefined names, but if you define an address with one of the predefined names, OMEGAMON for IMS uses the address that you define during the session.
- OMEGAMON for IMS places a comment character (>) in front of MDEF after it runs.

In the next example, MDEF gives the name XYZ to the address at offset *4A* in the CSA.

```
        MDEF  CSA+4A,XYZ
```

You can also use a predefined control block identifier to specify an anchor. OMEGAMON for IMS comes with predefined names for control blocks in a table. Use the MLST command to display the table.

Table 12. Partial list of valid control block identifiers

Identifier	**Description**
CVT	z/OS communications vector table.
ESCD	IMS extended system contents directory.
IMSTCB	Physical logger TCB.
SCD	IMS system contents directory.

Displaying and modifying storage

You can use authorized commands to display, scan, or modify address names for storage locations.

- "Storage display commands" on page 274
- "Storage scan commands" on page 277
- "Storage modification commands" on page 283

Storage display commands

You can use the storage display commands to display storage locations and to further analyze IMS control blocks and memory locations.

Some IMS education courses ask students to look through IMS control blocks in sample memory dumps. The `ILST` and `MLST` commands are important when issued with the `DUMP` command from the regular OMEGAMON for IMS command package. Students can use OMEGAMON for IMS commands to look at those control blocks in their own system while it is running.

ILST

Lists memory from the IMS address space.

ILST displays the contents of the private area of:

- the IMS control region
- the DBRC region
- the IRLM address space

Note: Some IMS private storage requires no authorization or special key to display: however, some areas are store-protected. To display these areas, you must supply the action character in the label field of the ILST command.

Type:

Immediate

Format:

aILSTcc addr,len

a

If required, an action character (-) in column 1.

cc

The format and source address space of the display.

p.

The first character specifies the display format:

b, B, or .

Hex and character (default).

C

Character only.

X

Hex only.

The second character specifies the address space OMEGAMON for IMS displays:

b or I

The IMS control region (default).

D

The IMS DLS region.

R

The IMS DBRC region.

L
The IMS IRLM region.

addr
The first address of storage that OMEGAMON for IMS displays. For more information, see “Address specification for storage commands” on page 272.

len
The number (up to eight hex digits) of bytes that OMEGAMON for IMS displays. Default is 16 (X’10’) bytes.

In the following example, ILST displays 80 (X’50’) bytes starting at hex address 15547C:

```
ILST  15547C,50
```

This command defaults to displaying the IMS control region in both hex and character formats as shown in Figure 170 on page 275.

```
ILST  15547C,50                                                 ADDR=0015547C
+ 0000 00000020 00000050 001559E0 80155B00  *.......&...\..$.*
+ 0010 007AA990 00000000 00000000 00000000  *.:z.............*
+ 0020 00000000 00000000 00000000 00000000  *................*
+ 0030 0014BD00 00145920 00155120 0014AD00  *................*
+ 0040 01540000 00144920 00155120 00000000  *        ........*
```

Figure 170. Typical output for ILST

MLST
Displays bytes of memory from either the common area or the OMEGAMON for IMS private storage area.

Type:
Immediate

Format:
aMLSTc addr,dlen

a
A K in the label position requests a display that shows the fetch protection key for each virtual block in the range that is specified, and whether fetch protection is ON or OFF.

c
Specifies the format of the output. Possible values are as follows:

B or b
Hex and character (default).

C
Character only.

X
Hex only.

addr
The first address of storage that OMEGAMON for IMS displays. For more information, see“Address specification for storage commands” on page 272.

dlen
The number (up to eight hex digits) of bytes that OMEGAMON for IMS displays. The default is 16 (X’10’) bytes.

In the following display, MLST lists 32 (X’20’) bytes, starting at address 1EB0, in character format.

```
MLSTC 1EB0,20                                  Addr= 00001EB0
+0000   *.........h.... .....167........*
```

Figure 171. MLST 1EB0 display

In the next display, MLST lists 16 (X’10’) bytes, starting at address FF32D6, in both hex and character formats.

```
MLST  FF32C1+15,10                              Addr= 00FF32D6
+0000  20280010 A18800F9 82F000FF 20400000   *....*h.9b0... ..*
```

Figure 172. MLST FF32D6 display

The next example shows MLST with a *K* in the label field, which displays fetch protection.

```
KMLST 7EF000,4000                               Addr= 007EF000
+      Virtual Block Number: 07EF000   KEY:5  Fetch Protection:OFF
+      Virtual Block Number: 07F0000   KEY:1  Fetch Protection:ON
+      Virtual Block Number: 07F1000   KEY:1  Fetch Protection: OFF
+      Virtual Block Number: 07F2000   KEY:0  Fetch Protection:OFF
```

Figure 173. KMLST display

XMLS

Displays storage from IMS address spaces.

Type:

Immediate

Format:

aXMLSc targ,addr,dlen

a

An action character in column 1.

-

Changes to a comment character (>) after the command runs.

<

Does not change to a comment character after the command runs. Use this action character to repeat the command.

c

The format of the output. Possible values are as follows:

B or b

Hex and character (default).

C

Character only.

X

Hex only.

targ

The IMS region. It can be:

$

Expands to the CICS currently monitored by OMEGAMON for IMS.

nnnn

Decimal ASID number.

cccccccc

Jobname.

addr

The first address of storage that OMEGAMON for IMS displays. For more information, see “Address specification for storage commands” on page 272.

dlen

The number (up to eight hex digits) of bytes that OMEGAMON for IMS displays. The default is 16 (X’10’) bytes. The maximum is 4096 (X’1000’) bytes.

The next three figures show uses of the XMLS command.

In Figure 174 on page 277, XMLS displays 32 (X’20’) bytes from address space 21 (starting at address 1EB0) in character format.

```
>XMLSC  21,1EB0,20
>storage at 00001EB0 in CICSTEST ASID=21:
> 0000 *ABCDEFGH IJKLMNOP QRSTUVWX Z0123456
```

Figure 174. XLMS output in character format

In Figure 175 on page 277, XMLS displays 16 (X’10’) bytes from the IMS region that is specified by the jobname USER14, starting at 1EB0, in both hex and character formats.

```
>XMLS  USER14,1EB0,10
>storage at 00001EB0 in USER14 ASID=21:
> 0000  C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7    *ABCDEFGHIJKLMNOP*
```

Figure 175. XLMS output in hex and character formats

In Figure 176 on page 277, XMLS displays 16 (X’10’) bytes from address space 21, starting at FF32D6, in hex and character format. The less-than symbol (<) prevents OMEGAMON for IMS from commenting out the command.

```
<XMLSB  21,FF32C1+15,10
+storage at 00FF32D6 in TSOA ASID=21:
+ 0000  4AA800F7 D3700000 00000000 000000F8      *y..7L..........8*
```

Figure 176. Output from XMLS command with an action character

The MPDD minor of the MAP1 command generates an XMLS command. OMEGAMON for IMS still requires password authority.

Storage scan commands

Issue the storage scan commands to scan storage locations for a specified string of values and display the string if it is found.

ISCN

Scans memory in the IMS address space.

Type:

Immediate

ISCN scans storage for a string of hexadecimal or character values. If the scan is successful,OMEGAMON for IMS displays the string.

ISCN scans the private areas of:

- IMS control region
- DLS region
- DBRC region
- IRLM address space

Note: Some IMS private storage requires no authorization or special key to scan; however, some areas are store-protected. To scan these areas, you must supply the action character in the label field of the ISCN command.

Format

aISCNcc addr,string,slen,dlen

a

If required, an action character (- or <) in column 1.

cc
The format and source address space of the display. Possible values are:

b or I
The IMS control region in hex and character format (default).

C
The IMS control region in character format only.

X
The IMS control region in hex format only.

.D
The IMS DLS region in hex and character format.

.R
The IMS DBRC region in hex and character format.

.L
The IMS IRLM region in hex and character format.

addr
The first address of storage that OMEGAMON for IMS scans. For more information, see "Address specification for storage commands" on page 272.

string
The hex string OMEGAMON for IMS uses for the scan. If you enclose it in single quotes, OMEGAMON for IMS assumes that it is a character string.

Note: OMEGAMON for IMS interprets two single quotation marks ('') within a character string as a single quotation mark (').

slen
The number (up to eight hex digits) of bytes that OMEGAMON for IMS scans. Default is 256 (X'100') bytes.

dlen
The number (up to eight hex digits) of bytes that OMEGAMON for IMS displays if the scan is successful. The display starts at the beginning of string. Default is 16 (X'10') bytes.

In the following example, ISCN scans 1280 (X'500') bytes of storage, starting at hex address 155000, for the hex string 00000020 and displays 80 (X'50') bytes starting at that point. This command defaults to displaying the IMS control region in both hex and character formats.

```
            ISCN     155000,00000020,500,50
```

The following figure shows typical ISCN command output if the string is found:

```
ISCN 155000,00000020,500,50                                         0015547C
+ 047C 00000020 00000050 001559E0 80155B00  *.......&...\..$.*
+ 048C 007AA990 00000000 00000000 00000000  *.:z.............*
+ 049C 00000000 00000000 00000000 00000000  *................*
+ 04AC 0014BD00 00145920 00155120 0014AD00  *................*
+ 04BC 01540000 00144920 00155120 00000000  *        ........*
```

Figure 177. ISCN command display

MCHN
Scans tables in either the common area or the OMEGAMON for IMS private storage area for a specified string of data.

Type
Immediate

MCHN scans the elements of a table for a string of hex or character values. If the scan is successful,OMEGAMON for IMS displays the table element that contains the string.

Use MCHN to examine the following components:

- Common Storage Area (CSA)
- System Queue Area (SQA)
- Nucleus

If you want to search private storage areas other than OMEGAMON for IMS's, use the XMCH command.

Format:
MCHNc addr,string,olen,chain,dlen

c
The format of the output. Possible values are:

B or b
Hex and character (default).

C
Character only.

X
Hex only.

addr
The address of the first table element that OMEGAMON for IMS scans. For more information, see "Address specification for storage commands" on page 272.

string
The hex string OMEGAMON for IMS uses for the scan. If you enclose this argument in single quotation marks, OMEGAMON for IMS assumes that it is a character string.

Note: OMEGAMON for IMS interprets two single quotation marks ('.') within a character string as a single quote (').

olen
The offset (in hex bytes) to the string in the table element; the comparison starts. You can precede *olen* with a plus sign (+) or a minus sign (-).

chain
The offset (in hex bytes) to the chain pointer (the location in the table element that contains the address of the next table element). You might precede *chain* with a plus sign (+) or minus sign (-).

dlen
The number of bytes (up to eight hex digits) that OMEGAMON for IMS displays if the scan is successful. The display starts at the beginning of the string. The default is 16 (X'10') bytes.

You must make sure that *addr* is the starting point of a table element. The address at *addr* + *chain* points to the next table element. The scan ends when the value at *addr* + *chain* is one of the following values: 0, -1, or *addr* (the table is a ring).

The following example shows a typical MCHN command.

```
        MCHN AAB6C8,D6C30199,8,4
```

In this example, MCHN scans a table that starts at location AAB6C8 and looks for the string D6C30199 that begins at the eighth byte of the table element: the address of the next table element is at offset 4. By default, this command displays 16 bytes of the table element in hex and character notation.

The following output displays if the scan is successful.

```
Addr=007DA000
 MCHN AAB6C8,D6C30199,8,4
+ 0000  E2E2C3E3 00000000 D6C30199 00000000  *SSCT    OC r    *
```

Figure 178. MCHN command display

MSCN

Scans storage for a string of data and displays the location.

MSCN scans the address space in which OMEGAMON for IMS stored for a string of hex or character values. If the scan is successful, OMEGAMON for IMS displays the string.

Type:
Immediate

Format:
MSCNc addr,string,slen,dlen

c
The format of the output. Possible values are:

B or b
Hex and character (default).

C
Character only.

X
Hex only.

addr
The first address of storage that OMEGAMON for IMS scans. For more information, see“Address specification for storage commands” on page 272.

string
The hex string OMEGAMON for IMS uses for the scan. If you enclose it in single quotation marks, OMEGAMON for IMS assumes that it is a character string.

Note: OMEGAMON for IMS interprets two single quotation marks ('') within a character string as a single quotation mark (').

slen
The number (up to eight hex digits) of bytes that OMEGAMON scans. The default is 256 (X'100') bytes.

dlen
The number (up to eight hex digits) of bytes that OMEGAMON displays if the scan is successful. The display starts at the beginning of the string. The default is 16 (X'10') bytes.

In the next example, MSCN scans the first 1000 bytes of the TIOT entry for the character string OIHELP (see “Address specification for storage commands” on page 272 for an explanation of the addressing), and displays 14 hex bytes starting at that point. The display is in both hex and character formats.

```
       MSCN 10%%+4%+C%,’OIHELP’,1000,14
```

Typical output of the MSCN command is shown as follows:

```
MSCN 10%%+4%+C%,’OIHELP’,1000,14                                 Addr=0061701C
+ 0000  D6C3C8C5 D3D74040 60BCA000 80001B00  *OIHELP  -.......*
+ 0010  14010100                             *....            *
```

Figure 179. MSCN command display

XMCH

Scans tables in the IMS address space.

XMCH scans the elements of a table for a string of hex or character values. If the scan is successful,OMEGAMON for IMS displays the table element that contains the string.

Use XMCH to search IMS address spaces. Use MCHN to search the address space in which OMEGAMON for IMS is stored.

XMCH scans the elements of a table in an IMS region for a string of hex or character values. If the scan is successful, OMEGAMON for IMS displays the table element that contains the string.

Use XMCH to search IMS regions. Use MCHN to search the address space in which OMEGAMON for IMS is stored.

Type:
Immediate

Format:
aXMCHc targ,addr,string,olen,chain,dlen

a
A required action character in column 1.

-
Changes to a comment character (>) after command runs.

<
Does not change to a comment character after command runs. Use this action character to repeat the command.

c
The format of the output. Possible values are as follows:

B or b
Hex and character (default).

C
Character only.

X
Hex only.

targ
The IMS region. It can be:

nnnn
Decimal ASID number.

cccccccc
Jobname.

addr
The address of the first table element that OMEGAMON for IMS scans. For more information, see "Address specification for storage commands" on page 272.

string
The hex string OMEGAMON for IMS uses for the scan. If you enclose it in single quotation marks, OMEGAMON for IMS assumes that it is a character string.

Note: OMEGAMON for IMS interprets two single quotation marks ('') within a character string as a single quotation mark (').

olen
The offset (in hex bytes) to the string in the table element; the comparison starts. You can precede *olen* with a plus sign (+) or minus sign (-).

chain
The offset (in hex bytes) to the chain pointer (the location in the table element that contains the address of the next table element). You might precede *chain* with a plus sign (+) or minus sign (-).

dlen
The number of bytes (up to eight hex digits) that OMEGAMON for IMS displays if the scan is successful. The display starts at the beginning of the table element. The default is 16 (X'10') bytes.

You must make sure that *addr* is the starting point of a table element. The address at *addr +c hain* points to the next table element. The scan ends when the value at *addr + chain* is one of the following values: 0, -1,*addr* (the table is a ring).

The following example shows a typical XMCH command.

```
        -XMCH USER14,AAB6C8,D6C30199,8,4
```

In this example, XMCH scans a table in the IMS region that starts at location AAB6C8 and looks for the string D6C30199 that begins at the eighth byte of the table element. The address of the next table element is at offset 4. By default, this command displays 16 bytes of the table element in hex and character format.

The following output displays if the scan is successful.

```
>XMCH USER14,AAB6C8,D6C30199,8,4
>Storage at 007DA000 in USER14 ASID=21:
> 0000  E2E2C3E3 00000000 D6C30199 00000000  *SSCT    OC r    *
```

XMSC

Scans storage in the IMS address space for a string of data and displays the location.

XMSC scans an IMS region for a string of hex or character values. If the scan is successful, OMEGAMON for IMS displays the string.

Type:
Immediate

Format:
aXMSCc targ,addr,string,slen,dlen

a
A required action character in column 1.

-
Changes to a comment character (>) after command runs.

<
Does not change to a comment character after command runs. Use this action character to repeat the command.

c
The format of the output. Possible values are as follows:

B or b
Hex and character (default).

C
Character only.

X
Hex only.

targ
The IMS region. It can be:

nnnn
Decimal ASID number.

cccccccc
Jobname.

addr
The address of the first table element that OMEGAMON for IMS scans. For more information, see “Address specification for storage commands” on page 272.

string
The hex string OMEGAMON for IMS uses for the scan. If you enclose it in single quotation marks, OMEGAMON for IMS assumes that it is a character string.

Note: OMEGAMON for IMS interprets two single quotation marks ('') within a character string as a single quotation mark (').

slen
The number (up to eight hex digits) of bytes that OMEGAMON for IMS scans. The default is 256 (X'100') bytes.

dlen
The number (up to eight hex digits) of bytes that OMEGAMON for IMS displays if the scan is successful. The display starts at the beginning of the string. The default is 16 (X'10')bytes.

In the next example, XMSC scans 1000 bytes in the IMS region that is specified by the jobname USER14, starting at location 515988 for the character string WORKAREA. If the scan is successful, OMEGAMON for IMS displays 14 hex bytes in hex and character format, starting at WORKAREA.

```
        -XMSC  USER14,515988,'WORKAREA',1000,14
```

The following output displays if the scan is successful.

```
>XMSC  USER14,515988,'WORKAREA',1000,14
>storage at 00515988 in USER14 ASID=21:
> 1B8 E6D6D9D2 C1D9C5C1 000000000 000C0000 *WORKAREA........*
> 1C8 000C002C *....*
```

Figure 180. XMSC command display

Storage modification commands

Use the storage modification commands to modify storage locations. Use these commands with caution.

IZAP
Zaps memory in the IMS address space

Type.
Immediate

CAUTION: IZAP is powerful. Use it with extreme care.

IZAP modifies the contents of the private areas of:

- The IMS control region
- The DLS region
- The DBRC region
- The IRLM address space

Consider the following points:

- Some IMS private storage requires no authorization or special key to modify; however, some areas are store-protected. To modify these areas, you must supply the action character in the label field of the IZAP command.
- If you use IZAP to modify storage in the pageable link pack area (PLPA), IZAP automatically completes a long-term page-fix to pageable link pack area (PLPA) page-fix to ensure that the storage remains modified. If necessary, OMEGAMON for IMS displays this message:

```
               PAGE(S) FIXED
```

Format:
aIZAPcc addr,ver,rep

a
If required, an action character in column 1:

-
: Changes to a comment character (>) after the command runs.

<
: Does not change to a comment character after the command runs. Use this action character to repeat the command.

cc
: The address space OMEGAMON for IMS modifies. Possible values are as follows:

  (blank) or .I
  : The IMS control region (the default).

  .D
  : The IMS DLS region.

  .R
  : The IMS DBRC region.

  .L
  : The IMS IRLM region.

addr
: The address of the string OMEGAMON for IMS might modify. For more information, see "Address specification for storage commands" on page 272.

ver
: The verify string.

  OMEGAMON for IMS modifies storage only if OMEGAMON for IMS finds this string at *addr:*

rep
: The replacement string. If OMEGAMON for IMS finds *ver* at *addr:*, *rep* replaces *ver.*

  Note: The verify and replacement strings must be the same length.

In the next example, IZAP changes a fullword at location 6764 from X'0000000A' to X'00000064'.

```
        IZAP  6744+20,0000000A,00000064
```

In the following example, IZAP changes a byte at location B46E08 from X'80' to X'40'

```
       IZAP  B46DE8+20,80,40
```

The next figure is a typical output from a successful IZAP.

```
>IZAP  B46DE8+20,80,40
>       >> MEMORY ZAP SUCCESSFUL. <<
```

MZAP
: Modifies the contents of the common area or the OMEGAMON for IMS private storage area.

  Type:
  : Immediate

- MZAP modifies the contents of the common area:
- Common Storage Area (CSA)
- System Queue Area (SQA)
- Nucleus
- Nucleus of system

Consider the following points:

- Some commonly addressable storage requires no authorization or special key to modify; however, some areas are store-protected. To modify these areas, you must supply the action character in the label field of the MZAP command. You can also use MZAP to zap storage in the OMEGAMON for IMS address space for debugging purposes.

- MZAP modifies up to 32 bytes of storage at a time from the address space in which OMEGAMON for IMS for CICS is stored; it is used primarily for modifying the z/OS common area. The verify and replace code lengths must be the same.
- If you use MZAP to modify storage in the pageable link pack area (PLPA), MZAP automatically does a long-term page-fix to ensure that the storage remains modified. If necessary,OMEGAMON for IMS displays the following message:

```
            PAGE(S) FIXED
```

Format:

-MZAP addr,ver,rep:

-

A required action character in column 1. The hyphen changes to a comment character (>) after the command runs.

addr

The address of the string that OMEGAMON for IMS might modify. For more information, see "Address specification for storage commands" on page 272.

ver

The verify string. OMEGAMON for IMS modifies storage only if OMEGAMON for IMS finds this string at *addr.* If OMEGAMON for IMS does not find the string, it displays what is at *add*.

rep

The replacement string. If OMEGAMON for IMS finds *ver* at *addr*,*rep* replaces *ver.*

Note: The verify and replacement strings must be the same length.

In the next example, MZAP changes a fullword at location 6764 from X'A' to X'64'

```
    MZAP  6744+20,0000000A,00000064
```

The next example shows how MZAP changes X'FF' to X'00' at location EA65C0.

```
    MZAP  EA65C0,FF,00
```

XMZP

Modifies another user's private storage area.

XMZP modifies the contents of the indicated private storage area in the IMS region

Type:

Immediate

Format:

-XMZP targ,addr,ver,rep

-

A required action character in column 1. The hyphen changes to a comment character (>) after the command runs.

targ

The IMS region. It can be:

cccccccc

Decimal ASID number.

nnnn

Jobname.

addr

The address of the string that OMEGAMON for IMS might modify. For more information, see "Address specification for storage commands" on page 272.

ver
The verify string. OMEGAMON for IMS modifies storage only if OMEGAMON for IMS finds this string at *addr*. If OMEGAMON for IMS does not find the string, it displays what is at *addr*.

rep
The replacement string. If OMEGAMON for IMS finds *ver* at *addr*, *rep* replaces *ver.*

Note: The verify and replacement strings must be the same length.

In the next example, XMZP changes a byte at location A0160 in the master scheduler address space.

```
        -XMZP  *MASTER*,A0160,0A,64
```

In the next example, XMZP changes a fullword in the target CICS address space at location 7088 from X'A' to X'64'.

```
        -XMZP  $,7088,0000000A,00000064
```

In the next example, XMZP changes a halfword at C4834 in the PRODJOB address space from X'1854'X'0700'.

```
        -XMZP  PRODJOB,C4834,1854,0700
```

In the example that follows, XMZP changes a halfword at location D7E30 from X'18C0' to X'18C1'.

```
        -XMZP  BMP01,D7E30,18C0,18C1
```

Typical output (if the zap is successful) is as follows:

```
.XMPZP      BMP01,D7E30,18C0,18C1
>               >>Memory Zap Successful<<
```

IMS MTO console support

You can issue IMS commands or transactions from the OMEGAMON for IMS console and display IMS messages and IMS-related z/OS messages at the OMEGAMON for IMS console by using the IMS MTO console support commands.

- "IMS commands and transactions" on page 286
- "IMS messages and IMS-related z/OS messages (ICNS)" on page 287

IMS commands and transactions

You can use the `ICMD` command to enter any IMS command or transaction.

ICMD
Enters an IMS command or transaction.

Type:
Immediate

Format:
-ICMD [ssid] cc...cc

-
OMEGAMON for IMS requires the action character (-) in column 1 to run the ICMD command.

[ssid]
Where [ssid] is optional and applies only to IMS DBCTL environments. The ssid is the subsystem identifier of the DBCTL system in which to run this command or transaction.

cc...cc
Any string that you can enter at an z/OS console in response to the IMS READY WTOR. ICMD is much more convenient to use because you do not have to know the WTOR reply ID number.

The next three figures are typical ICMD commands.

```
-ICMD  /DIS POOL CIOP
```

Note: In the following example, a period ends the transaction.

```
-ICMD  PART 8734532.
```

```
-ICMD  /DIS POOL CIOP
```

The ICMD command uses the WTO macro to echo commands you enter; this echo displays on the ICNS display. (See "IMS messages and IMS-related z/OS messages (ICNS)" on page 287.) Because OMEGAMON for IMS uses the IMS WTOR, you must end the IMS transaction message with a period (.), or IMS thinks you only entered the first segment of a multi-segment input message, and prompts you to enter the next segment. The z/OS (WTOR) console, and therefore the ICMD command always has the same authorization as the IMS master terminal. See Figure 181 on page 289 for an example. The output also displays on the z/OS master console because ICMD uses IMS's WTOR console support. OMEGAMON for IMS passes the text that you enter by using the ICMD command to IMS by REPLYing to the outstanding IMS WTOR (IMS READY message). In response, IMS displays any reply by using WTO.

OMEGAMON for IMS runs only one ICMD command per cycle: OMEGAMON for IMS defers any other ICMD commands on the screen until later cycles. This happens because IMS has only one WTOR available.

If two users at two different OMEGAMON for IMS terminals try to issue the ICMD command at the same instant, it is possible (although unlikely) for OMEGAMON for IMS to run only one of the commands. This happens because, for a short interval, there is no IMS reply ID to respond to. The ICMD command displays the following message

```
        >> IMS Reply ID not found: RC=4 <<
```

In this case, issue the ICMD command again. If the ICMD command fails repeatedly or displays a return code other than 4, contact IBM Software Support.

The ICMD command uses an SRB to extract the current IMS reply ID number. The resource usage is not significant, but "The SRB/XMS facility" on page 271 describes the operational factors that apply in this case.

IMS messages and IMS-related z/OS messages (ICNS)

You can use the ICNS command to display the WTO and MTO message traffic that the IMS control regions and its dependent regions produce. These messages include any responses to commands and transactions that you enter by issuing the ICMD command.

OMEGAMON for IMS does not include DFS996I IMS/VS READY messages on the display, since these have no value. Similarly, OMEGAMON for IMS removes the prefix DFS000I from informational messages that begin with this identifier.

ICNS shows a consolidated display of IMS and z/OS messages that are associated with IMS control and dependent regions. The command has a buffer to retain 99 messages for display in chronological order. OMEGAMON for IMS only generates null replies if the ICNS command remains on the OMEGAMON for IMS

During JES hotstarts, ICNS displays the following message:

```
       ICNS unable to receive messages from JES at this time
```

This message displays temporarily until the hotstart is complete, when ICNS resumes normal functioning. Every time an IMS region issues a WTO, ICNS captures and stores the message in a buffer accessible to OMEGAMON for IMS sessions. (ICNS does not disturb the messages; they still display on z/OS consoles as usual.) Use the buffer to display as many as the most recent 99 WTO messages. It also primes the buffer with a message that provide the date and time the support is activated.

ICNS

Displays IMS WTO and MTO message traffic.

Type:

Immediate

Format:

-ICNSnn

-

A required action character (-) in column 1.

nn

The number (01-99) of lines OMEGAMON for IMS displays. ICNS displays the last *nn* lines of intercepted messages.

Note: Since ICNS is only a display command, it does not automatically convert to a comment after one execution; ICNS can run as many times as you want it to.

```
        -ICNS18
```

You can use the ICNS command with the ICMD authorized command (see Figure 181 on page 289 for an example).

ICNS only captures messages from *active* IMS regions. For example, when a new message region or BMP starts, ICNS does not see the JOB STARTED messages. At the time JES issues those messages, IMS is not yet aware of the existence of the region. ICNS intercepts messages only if the issuing address space currently has an entry in the IMS SVC directory table; that is, if IMS initialized the address space and assigned it a PST.

Similarly, when a dependent region ends, ICNS does not see the JOB ENDED messages, since JES issued them after the region disassociated itself from IMS. Between startup and shutdown however, ICNS sees every message that displays on the JES job log of the control region or any of its dependents. (However, ICMD suppresses the DFS996I IMS READY WTOR messages. Since the ICMD authorized command automatically extracts the IMS reply ID number, the user does not need this message and ICNS removes it to eliminate clutter.)

ICNS observes messages that originate from the DLS, DBRC, IRLM, and Virtual Fetch address spaces.

ICNS completes another useful function with ICMD. While the output generated by IMS commands issued at the z/OS console displays immediately, output from transactions entered at the z/OS console and unsolicited (for example, application program ABEND) messages do not display on the z/OS console until you reply again to the next IMS WTOR message. (IBM's *IMS/VS Operator's Reference Manual* states that this function prevents important IMS messages from getting lost in the heavy traffic to the z/OS operator console.) Each OMEGAMON for IMS cycle, the ICNS command checks whether there are any queued messages for the WTOR LTERM. If there are, it automatically issues a null reply (R xx,' '), which forces the output to display on the z/OS console (and the ICNS display on the next cycle). This prevents you from having to issue the dummy reply yourself, or worse, not seeing an important IMS message until it is too late. The automatic reply feature is only active when ICNS is on the OMEGAMON for IMS screen.

The following figure shows a typical ICNS command that is used with the ICMD command.

```
================================================================================
>ICMD  /DIS ACTIVE                                                      RC =  0
================================================================================
-ICNS18  IMS Console Display
+    NMBUFS   9 VRDS      0 FOUND     0 VWTS      0  ERRORS 00/00   IPO1
+  DATA BASE BUFFER POOL: BSIZE 8192                        IPO1
+  RRBA        0 RKEY     0 BFALT     0 NREC      0 SYN PTS     0   IPO1
+    NMBUFS   9 VRDS      0 FOUND     0 VWTS      0  ERRORS 00/00   IPO1
+  DATA BASE BUFFER POOL: BSIZE  142848                             IPO1
+  RRBA        0 RKEY     0 BFALT     0 NREC      0 SYN PTS     0   IPO1
+    NMBUFS  45 VRDS      0 FOUND     0 VWTS      0  ERRORS 00/00   IPO1
+  *82160/150719*   IPO1
+  R 81,/DIS ACTIVE
+  REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  CLASS   IPO1
+        MSGRGN    TP    NONE   IPO1
+        BATCHREG  BMP   NONE   IPO1
+  VTAM ACB OPEN          -LOGONS ENABLED     IPO1
+  LINE ACTIVE-IN -   1 ACTIV-OUT -   0   IPO1
+  NODE ACTIVE-IN -   0 ACTIV-OUT -   0   IPO1
+  *82160/150724*   IPO1
================================================================================
```

Figure 181. ICNS command with the ICMD authorized command

z/OS operator console support

You can issue a z/OS or JES2 operator command and display a z/OS console buffer by using the z/OS operator console support commands.

OCMD

Issues z/OS and JES2 operator commands from an OMEGAMON for IMS terminal.

Type:

Immediate

Format:

-OCMDnn <CONS=<conid|conname>> ccccc

OMEGAMON for IMS issues the command (*ccccc*) that you provide through SVC 34.

OCMD requires the action character (-) in column 1. The action character changes to a comment character after execution.

For commands that accept a return destination, the variable *nn* indicates which operator's console receives the response. If you omit *nn*, the response goes to the master console.

The variable *ccccc* is a z/OS or JES2 command.

conid specifies which operator's console issued the command. This operand overrides the *nn* operand.

conname specifies the console name from which the command originated.

The following screen display shows an example of the OCMD command.

```
-OCMD01  SEND ‘PLEASE RELEASE ALL HELD DATA SETS’,USER=TSO001
-OCMD01  $HJ123
-OCMD01  C JOB123,DUMP
```

OMEGAMON for IMS has the same z/OS console authority as the console you indicate. Therefore, if you want to issue a command that requires master console authority (such as VARY CHANNEL) you must specify the z/OS console ID of your current master console. If you do not specify a console ID, the master console ID is used.

Any command you issue with the OCMD command displays the generated output on those z/OS consoles that respond to commands that system tasks issue. To specify which consoles are to receive the output, supply a two-digit operand to OCMD. This number is the UCMID (or internal z/OS identifier) of the console on which the z/OS response is displayed. For example,

```
        -OCMD04 D T
```

Displays the date and time on console number 04. This number ranges from 01 to the maximum number of consoles that are generated for your installation. To see the console number (UCMID) for each of your z/OS consoles, issue the D CONSOLES command at a real z/OS console.

To see the output of a z/OS command you issue, use the CONS authorized command.

RCMD

Routes z/OS and JES2 commands to a required system in a sysplex.

Type:
Major

Format:
-RCMDnn sysname <CONS=<conid|conname>> ccccc

OMEGAMON for IMS issues the command (*ccccc*) that you supply through SVC 34.

RCMD requires the action character (-) in column 1. The action character changes to a comment character after execution.

The variable *nn* indicates which operator's console receives the response. If you omit *nn*, the response goes to the master console.

sysname specifies the system name where the command runs.

The variable *ccccc* is a z/OS or JES2 command.

conid specifies which operator's console issued the command. This operand overrides the *nn* operand.

conname specifies the console name from which the command originated.

CONS

Displays the console image for the specified console.

Type:
Major

Format:
CONSnn <CONS=<conid|conname>>

The variable *nn* is a z/OS console number. This number ranges from 01 to the maximum number of consoles that are generated for your installation. If *nn* is omitted, OMEGAMON for IMS selects the master console.

conid selects the z/OS operator's console by ID number. This operand overrides the *nn* operand.

conname selects the z/OS operator's console by name.

CONS displays three types of information about the command line: the type of console (CONSOLE or MASTER CONSOLE), the console's device number, and the console's ID number (as in ID=3).

To see each z/OS console's number (also known as the UCMID), issue the D CONSOLES command at a real z/OS console.

Use the CONS minor commands to display selected types of messages from the console.

The CONS command also displays the output that z/OS or JES commands generate when the OCMD authorized command issues them. (See Figure 182 on page 291 for an example.)

The OMEGAMON for IMS product accesses the CONSOLE address space by using cross memory services each time you ask for a display. It uses a small amount of memory in the z/OS common area (approximately 2K) for the period that the CONS command is on the screen.

Figure 182 on page 291 is an example of an OMEGAMON for IMS screen that uses the z/OS console support.

```
>OCMD R 83,ABEND                                                        RC =  0
================================================================================
 CONS01  MASTER CONSOLE 01F
 line99   *STC  721 *82 DFS996I *IMS READY*  IPO1
+        - STC 1113  $HASP395 IMSRDR   ENDED
+          STC 1113  $HASP250 IMSRDR   IS PURGED
+        - JOB 1114  $HASP373 BMP01    STARTED - INIT  4 - CLASS I - SYS A430
+        - JOB 1111  $HASP395 TSO003G  ENDED
+                    $HASP309    INIT  3 INACTIVE ******** C=I
+          JOB 1111  $HASP250 TSO003G  IS PURGED
+        - STC  721  DFS2500I *MDA00   IPO1
+        - DATABASE BE3PARTS SUCCESSFULLY ALLOCATED
+        - STC  721  DFS2500I *MDA00   IPO1
+        - DATABASE BE3PSID1 SUCCESSFULLY ALLOCATED
+        | JOB 1114 @83 DFS3125A PRIMER SAMPLE TEST, REPLY CONT, LOOP, ABEND, O
+        |  CANCEL JOB
+          TSU 1115  $HASP100 TSO045   ON TSOMEGAMON/IMSNRDR
+        - TSU 1115  $HASP373 TSO045   STARTED
+      00            IEE600I REPLY TO 83 IS;ABEND
================================================================================
```

Figure 182. OCMD and CONS output

CONU

Locates the output buffer for a z/OS operator console by device address.

Type:
Major

Format:
CONU xxxx

Note: CONU Command now accepts input in either a three-digit or four-digit format.

CONU functions like CONS, except that you supply the three-byte hex device address as an operand instead of the console ID. If you omit *xxx*, OMEGAMON for IMS automatically locates the output buffer for the master console. CONU displays the address and the console ID as shown:

```
CONU  6A00   Master Console 6A00 ( ID=3 )
```

The following minor commands display information about the selected console.

ACTN

Displays only those lines from the screen that require some action.

Type:
Minor of CONS and CONU

ACTN displays any line that begins with an asterisk (*).You can use this command to display any outstanding Mount or Reply messages that need attention. ACTN has no arguments.

For example, the ACTN command displays the IMS WTOR message that is issued from the control region that OMEGAMON for IMS is monitoring. A typical action message command displays as follows:

```
 CONS03   Console 640 (ID=3)
 actn
+ 0100 08.26.21 STC 2188 *12 DFS996I *IMS READY* IMSD
```

Figure 183. ACTN command display

LINE

To display the last 12 lines of the screen of console 11, enter this command:

```
        CONS11   Master Console (ID=11)
        line12
```

Displays the last *nn* lines from the console you select.

Type:
Minor of CONS and CONU

Format:
LINEnn

The LINE minor command displays only in-line messages, such as the display produced by the following command.

```
        D A,L,L=Z
```

LINE does not display out-of-line messages, such as those produced by the following command.

```
        D A,L,L=A
```

To avoid this situation, either use the L=Z operand where appropriate, or issue the following console control command to remove the out-of-line display areas:

```
         K A,NONE
```

MNT
Displays mount messages that require operator action.

Type:
Minor of CONS and CONU

MNT has no arguments.

For example, to display mount messages for console 4, enter the following command:

```
 CONS04   Console 660 (ID=4)
 mnt
+ 2000 09.07.47 JOB 2225 *13 IEC701D M 370,VOLUME TO BE LABELED CA1759
```

Collecting data about address spaces

You can use the PEEK major command to collect information from the target address space, and use the minor commands to control the format and display of the information.

PEEK
Collects information about a single address space.

Type:
Major

Format:
aPEEK targ

a
An action character in column 1:

-
Specifies that new data be collected from the target address space.

<
Specifies that the command is re-executed on succeeding cycles.

b
Enables minor commands to run with previously collected data.

targ
The target address space. It can be:

cccccccc
jobname

nnnn
Decimal ASID number

*****
OMEGAMON for IMS address space

For example, to gather data from a job named PAYROLL (with an ASID of decimal 25), enter:

```
-PEEK PAYROLL
```

or

```
-PEEK 25
```

The PEEK command accesses the PAYROLL address space, removes the action character from column 1, and displays the following output:

```
PEEK  PAYROLL  ASID=25  >> OB8112:  Data Collection Initiated <<
```

When the PEEK command collects the data from the target address space and stores it in the work area, it displays this information:

```
PEEK  PAYROLL  ASID=25, collected at 15:39:39
```

Any PEEK minor commands that you issue now examine this work area. Each time that you issue the PEEK with the action character in the label field, the PEEK collects current information. If you issue the PEEK without an action character, it uses the data in the work area from the previous update.

The PEEK command uses cross memory services to access the target address space for a job that runs non-swappable. For a swappable job, it uses an SRB routine. Because SRBs run at the highest priority and increase the swapping load, overuse can degrade performance. The `.SET` command contains two keywords to set OMEGAMON for IMS profile parameters for the PEEK command. The keywords are LOOPCOUNT and PEEKSIZE.

LOOPCOUNT
Sets the maximum number of control blocks that the PEEK command tests before it detects a loop. The valid range is 1 - 60000.

The PEEK command traces control block chains. If OMEGAMON for IMS encounters a damaged target address space, some of the control blocks that are examined might have chained into a loop, and OMEGAMON for IMS issues a warning message. The warning might also display when you chain through an address space that has a complex TCB structure. In this case, the cause might not be a loop, but rather the large amount of processing that is necessary to scan all of the TCBs.

PEEKSIZE
Sets the work area size (in bytes) for the PEEK command. The maximum is 204800.

The first time that you use the PEEK command in an OMEGAMON for IMSsession, it obtains a work area (32K by default) from the private area to hold the collected data. OMEGAMON for IMS gives you a warning message if the data does not fit within the work area.

Issue the `.SET` command and increase the value of PEEKSIZE= to increase the work area size. Then, reissue the PEEK command and it collects the data. You can save the new PEEKSIZE definition in a user profile.

The following minor commands can be used with the PEEK command:

AMAP
Displays a map of virtual storage usage within the private area.

Type:
Minor of the PEEK command

The AMAP minor command of the PEEK command displays a map of virtual storage utilization within the private area. This map indicates the maximum region available, the portion currently in use, and various areas within the region.

The AMAP display for XA and ESA shows all virtual storage, or you can limit the map to storage above or below the 16M line with an A or B argument as shown.

```
PEEK  USER01   ASID=46, collected at 15:39:39
amap  <map all virtual storage>
amapA <map virtual storage above the 16M line> (XA and ESA)
amapB <map virtual storage below the 16M line>
```

Figure 184. AMAP command display

Figure 185 on page 294 shows the AMAP display format for XA and ESA

```
 PEEK  USER01   ASID=46, collected at 15:39:39
 amap
+      ===== 2 Gig Line ====== <== 7FFFFFFF Highest 31-bit address
+      |                     |
+      |---------------------| <== 7FFFFFFF  Top of Extended Private
+      | ////////////////// |
+      | //  System Area  // | >-----------   8,948K ELSQA,SWA unallocated
+      | ////////////////// |                     6K Fragmented free space
+      |---------------------| <== 7F741000  Current bottom of ELSQA,SWA
+      |                     |
+   A  |      Available      | >-----------   1,966M Avail. for ELSQA/SWA only
+   M  |                     |
+   A  |---------------------| <== 048FFFFF  Extended User Area Limit
+   P  |                     |
+   A  |      Available      | >-----------  32,668K Avail. for ELSQA/SWA/USER
+      |                     |
+      |---------------------| <== 02918FFF  Current Top of Ext. User Area
+      | ////////////////// |                     4K Largest free block
+      | ///  User Area  /// | >-----------       4K Extended User unallocated
+      | ////////////////// |                     7K Fragmented free space
+      |---------------------| <== 02900000  Bottom of Extended Private
+      |                     |
+      ===== 16 Meg Line ===== <== 00FFFFFF  Highest 24-bit address
+      |                     |
+      |---------------------| <== 007FFFFF  Top of Private
+      | ////////////////// |
+   A  | //  System Area  // | >-----------      24K LSQA/SWA unallocated
+   M  | ////////////////// |                    42K Fragmented free space
+   A  |---------------------| <== 007D0000  Current Bottom of LSQA/SWA
+   P  |                     |
+   B  |      Available      | >-----------   2,796K Avail. for LSQA/SWA only
+      |                     |
+      |---------------------| <== 00514FFF  User Area Limit
+      |                     |
+      |      Available      | >-----------     104K Avail. for LSQA/SWA/USER
+      |                     |
+      |---------------------| <== 004FAFFF  Current top of User Area
+      | ////////////////// |                   128K Largest free block
+      | ///  User Area  /// | >-----------     204K User unallocated
+      | ////////////////// |                    40K Fragmented free space
+      |---------------------| <== 00005000  Bottom of Private
+      === Absolute Bottom === <== 00000000  Prefixed Storage Area
```

Figure 185. AMAP display format (XA and ESA)

Area descriptions for above the 16M line:

Highest 31-bit address
The highest possible address in 31-bit architecture.

Top of extended private
Highest address within the extended private area.

ELSQA,SWA unallocated
The amount of storage that is not currently allocated within the extended system area.

Fragmented free space
The amount of free storage within allocated pages of the extended system area.

Current bottom of ELSQA,SWA
: Lowest address that is allocated within the extended private area for the extended system area.

Avail. for ELSQA/SWA only
: The amount of deallocated storage between the current bottom of the extended system area and the limit of the extended user area.

Extended User Area Limit
: Highest address possible for the extended user area.

Avail. for ELSQA/SWA/USER
: The amount of unallocated storage between the extended user area limit and the current top of extended user area. The extended system area can allocate storage within this area.

Current Top of Ext. User Area
: The highest address that is currently allocated within the extended private area for the extended user area.

Largest free block
: The largest contiguous piece of deallocated storage within the extended user area.

Extended User unallocated
: The amount of storage not allocated within the extended user area.

Fragmented free space
: The amount of free storage within allocated pages of the extended user area.

Bottom of Extended Private
: The lowest address that is currently allocated within the extended private area for the extended user area.

Area descriptions for below the 16M line:

Highest 24-bit address
: The highest possible address in 24-bit architecture.

Top of Private
: Highest address below the common area (start of CSA).

LSQA/SWA unallocated
: Total of contiguous 4K areas. The numbers include LSQA, SWA, and subpools 229/230.

Fragmented free space
: Total of areas within LSQA that are each less than the 4K available for allocation as defined by FQEs.

Current bottom of LSQA/SWA
: Lowest address that is allocated to LSQA/SWA subpools.

Avail. for LSQA/SWA only
: Total space available for LSQA/SWA allocation, which includes the LSQA/SWA deallocated value and the amount of space in the region available area.

User Area Limit
: Highest address available for user allocation (region size plus IEALIMIT).

Avail. for LSQA/SWA/USER
: Amount of space available for problem program allocations, not including deallocated areas within the region used.

Current top of User Area
: Highest address that is currently allocated for problem program use.

Largest free block
: Largest contiguous area available within the region used.

User unallocated
: Total of the contiguous 4K areas within the region that is used which are available for problem program use.

Bottom of Private
Lowest address within the private area (above the resident nucleus rounded up to the next 64K boundary).

Prefixed Storage Area
Fixed storage location that start with absolute zero.

DATA
Displays data space and Hiperspace utilization for an address space.

Type:
Minor of the PEEK command

The DATA minor of the PEEK command requires an APF-authorized environment. For data space information display, z/OS/SP™ 3.1 (ESA) must be installed. For Hiperspace™ information display, DFP 3 must be installed.

Here is an example of the DATA minor display.

```
-PEEK TSOX07   ASID=48   >> OB8112: Data Collection Initiated <<
 data   Name    Type        Owning Task     Current Size  Maximum Size
+     SDUMPSWA  Basic   CR8SPACE (007FE380)        64K          256K
+     MYHIPER   Scroll  HIPERPGM (007FD468)      1000K         4096K
+     HIPER2    Cache   HIPERPGM (007FDA50)        40K          100K
+     SDUMPALL  Basic   DSPPGM1  (007ED900)         4K            4K
```

Figure 186. DATA minor display

If you put an *X* in the label field, DATA displays extended information as shown in the following figure.

```
-PEEK TSOX07   ASID=48   >> OB8112: Data Collection Initiated <<
Xdata   Name    Type        Owning Task     Current Size  Maximum Size
+     SDUMPSWA  Basic   CR8SPACE (007FE380)        64K          256K
+               Key= 8  Fprot=NO   Dref=NO   Scope=SINGLE
+     MYHIPER   Scroll  HIPERPGM (007FD468)      1000K         4096K
+               Key= 7  Fprot=NO
+     HIPER2    Cache   HIPERPGM (007FDA50)        40K          100K
+               Key= 0  Fprot=YES  Castout=YES
+     SDUMPALL  Basic   DSPPGM1  (007ED900)         4K            4K
+               Key= 8  Fprot=NO   Dref=NO   Scope=ALL
```

Figure 187. XDATA display

Name
Name of the data-only space.

Type
Type of data-only space. Valid types are basic, scroll, and cache.

BASIC
BASIC data space.

SCROLL
SCROLL-type Hiperspace.

CACHE
CACHE-type Hiperspace.

Owning Task
Program name and TCB address that is associated with the owning task.

Size
Current size of the data-only space in K.

Max Size
Maximum allowable size of the data-only space in K.

Key
Storage protect key of the data-only space.

Fprot
Storage fetch protection indicator of the data-only space.

Dref
Disabled reference storage indicator for the BASIC data space.

Scope
Specifies whether the BASIC data space is shareable (ALL) or non-shareable (SINGLE) with other address spaces.

Castout
Indicates whether the CACHE-type Hiperspace is given special consideration when the system searches for pages to remove from expanded storage when a shortage arises.

DDNS
Displays information about allocated ddnames.

Type:
Minor of PEEK

Format:
[␣]DDNS[nn][X]DDNS[nn]

␣
Displays all ddnames that are allocated to a jobstep and their corresponding device addresses, data set names, and volume serial numbers.

X
Requests extended information. For each TCB group of ddnames, XDDNS shows:

LRECL
Logical record.

BLKSZ
Blocksize.

RECFM
Record format.

DSORG
Data set organization.

PWD
Password protection. A blank indicates that no data set password is in effect.

EXCP
Run channel program. This figure represents the number of I/Os (EXCPs) issued.

TIOT
Task I/O table address.

nn
Suppresses the first *nn* lines of the display. This option is useful if all of the ddnames do not fit on one screen.

Here is an example of the DDNS display:

```
 PEEK  USER01   ASID=46, collected at 15:39:39
 ddns   DDname  Adr VolSer Sta,Dsp DSname
+      PROC00   245 SYSRES SHR,KEE SYS1.PROCLIB
+               246 USER01 SHR,KEE USER1.PROCLIB
+      PROC01   245 SYSRES SHR,KEE SYS2.PROCLIB
```

Figure 188. DDNS display

The following conditions apply to the DDNS display:

- The DDNS display includes dynamically allocated ddnames.
- A blank ddname field indicates the data set is part of a concatenation with preceding data sets.

- DDNS displays only the first allocated volume in cases where a ddname is associated with a multi-volume data set. A plus sign (+) immediately following the volser field indicates a multi-volume data set.

JOBS

Displays values available from the private area.

Type:

Minor of PEEK

The following screen shows a typical JOBS display.

```
 PEEK  XOIDI    ASID=25, collected at 15:39:39
 jobs  Job started at  7:58:26 on 12/29/99  JOBCLASS=F  MSGCLASS=D
+      Programmer Name=JOHN SMITH
```

Figure 189. JOBS display

When the **Programmer Name** field is blank in the job statement, the JOBS minor command does not display that line. JOBCLASS information displays only for batch jobs. The job class is that class that is originally associated with the job. If a JES command is requeued before execution, the original job class from the CLASS= parameter in the JCL JOB statement is displayed.

MODS

Displays information about the modules that are currently loaded into the jobpack area of the user.

Type:

Minor of PEEK

The following screen display shows a typical example of the MODS command.

```
PEEK  USER01   ASID=46, collected at 15:39:39
 mods
+      Module    Entry     Length  Users  Attributes
+      AUTHMOD   00073204  02FAF8      1  RENT,REUS,AUTH,AC=1
+      PAY1      0003D140  032D90      0  RENT,REUS,AUTH,AC=1,ALIAS(PAYMOD01)
+      ERROR     000A7128  000890      1  RENT,REUS
+      ISPTCM    000093E8  000418      1  RENT,REUS,AUTH
```

Figure 190. MODS command display

The name of each module currently in the user’s jobpack area displays along with its entry point address, length, use count, and load module attributes. For the definition of the attributes, see the *IBM z/OS Linkage Editor Manual*.) You can use the entry point address with the cross-memory list or zap (XMLS or XMZP) commands.

To suppress the display of the first *nn* or *nnn* modules, you can optionally specify a two- or three-digit number in the operand field of MODS (MODS*nn* or MOD*nnn*). This is useful if all of the names do not fit on one screen.

STEP

Displays private area storage utilization.

Type:

Minor of the PEEK command

The following screen shows an example of the STEP command for z/OS/XA and z/OS.

```
 PEEK  USER01   ASID=46, collected at 15:39:39
 step  Job Step Pgm: cccccccc; 5 TCBs, 3 datasets, and 3 Modules
+      Step started at 18:29:04 , now in step # 1 of 1
+
+ For the region below the 16M line:
+
+   Total private region = 8172K Unused = 3104K
+   Region requested = 5120K Region limit = 5184K
+   Low PVT in use = 4876K Unallocated = 204K Free = 40K
+   High PVT in use = 192K Unallocated = 24K Free = 42K
+   Start of SYSREG:   00001000  End of SYSREG:   00004FFF
+   Start of low PVT:  00005000  End of low PVT:  004FAFFF
+   Current top:       004FAFFF  Limit of region: 00514FFF
+   Start of high PVT: 007D0000  End of high PVT: 007FFFFF
+
+ For the extended region above the 16M line:
+
+   Total private region = 2055168K Unused = 2046116K
+   Region limit = 32768K
+   Low PVT in use = 96K Unallocated = 4K Free = 7K
+   High PVT in use = 8956K Unallocated = 8948K Free = 6K
+   Start of low PVT:  02900000  End of low PVT:  02918FFF
+   Current top:       02918FFF  Limit of region: 048FFFFF
+   Start of high PVT: 7F741000  End of high PVT: 7FFFFFFF
===========================================================================
```

Figure 191. STEP command for z/OS/XA and z/OS

cccccccc
The current program name of the topmost JOBSTEP TCB. (This value corresponds to what is in the EXEC statement, unless XCTL is used to transfer control to another load module.) The other counts indicate how many lines of output you might expect from the TCBS, DDNS, and MODS minor commands.

Total private region
The total size of the private area, including areas that cannot be allocated.

Region requested
The amount that you specify on the REGION JCL parameter.

Region limit
The region limit that the IEALIMIT exit imposes.

Low PVT
The storage that the REGION parameter limits, which include all of the user subpools.

High PVT
Includes LSQA, SWA, and subpools 229 and 230. This value is allocated from the top of the user's region downward and is not limited by the REGION JCL parameter.

In use
The storage that is allocated to subpools.

Unallocated
The storage that is not allocated to subpools.

Free
The storage that is allocated to subpools but not currently GETMAINed.

SYSREG
An area of storage that is reserved for use by the region control task.

Current top
The highest allocated address in the low PVT area at the current time.

SUBP
Displays information about current virtual storage allocations for each storage subpool.

Type:
Minor of PEEK

Format:
[␣]SUBP[nn][X]SUBP[nn]

b
: Requests information for the subpools that are allocated to the jobstep TCB and to any TCBs that follow it.

X
: Requests subpool information for *all* TCBs in the address space.

nn
: Suppresses the first *nn* lines of display. Since the SUBP display is likely to exceed the size of the physical screen, this option is provided to limit the display. For example, entering SUBP10 suppresses the first 10 subpools in the display.

The SUBP display consists of two parts for each TCB: one part shows detailed information about the allocation of storage and one part shows a summary of the virtual storage. The third section shows totals. Notice that the totals displayed include all subpools in the address space, whether the X label is entered in front of the SUBP command. Consequently, the totals might be larger than the sums of the private area statistics.

The SUBP display shows only allocation statistics for private area subpools and not common area subpools such as Subpool 241.

Here is an example of the display (for both XA and ESA).

```
 PEEK  CPSTEP6A ASID=15, collected at 15:39:39
 subp
+ SBP-K   Alloc  Real #Blks      Addr    Free  #Blks  Mxfree   Program
+ 251-8     28K   28K    10  000DD000    6184      3  000968   IFOX&zz    Own
+   0-8   2012K   22k     6  000E4000    5608      3  000AF0             Shr
+ 230-5      4K    1K     1  007CA000    3966      5  000488             Own
+ 237-1    188K   41K    39  007CC000    3156     38  000440             Shr
-----------------------------------------------------------------------------
+ PVT-Hi:    4K    1K     1                3K      5
+ PVT-Lo:   28K    7K     3                4K      3
+ Subtot:   32K    8K     4                7K      8
==============================================================================
-----------------------------------------------------------------------------
+ Tot-Hi:  360K   78K    70             40K     68
+ Tot-Lo: 2044K   29K     8             13K      7
+ Totals: 2404K  107K    78             53K     75
==============================================================================
```

Figure 192. SUBP display

The first portion of the display for each TCB shows detailed information about the subpool:

SBP-K
: Subpool number and protect key (decimal).

Alloc
: Amount of virtual storage that is currently allocated to the subpool (in 4K increments). The storage is not necessarily contiguous.

Real
: Amount of real storage that backs up the virtual allocation.

#Blks
: Number blocks allocated to the subpool.

Addr
: Address of the block with the lowest address (hex). In this example, there are 10 blocks and the lowest starts at 000DD000 in subpool 251.

Free
: Number of free bytes (hex) within the subpool that no one has yet GETMAINed. Anything in a free area is available for a GETMAIN for the same subpool, but not for other subpools.

#Blks
: Number of free non-contiguous blocks within the subpool, where each block can be any number of bytes (in 8-byte units).

Mxfree
Size (hex) of the largest free block within the subpool.

Pgmname
Program name of the TCB that is associated with these subpools.

Own|Shr
Allocations marked SHR are displayed for each TCB that shares the allocation. Allocations marked OWN display only for the owning TCB.

The next portion of the display for each TCB is a summary of the virtual storage allocated.

PVT-Hi
Summary of LSQA allocated for the address space.

PVT-Lo
Summary of user storage that is allocated for the address space.

Subtot
Summary of LSQA and user virtual storage allocated.

The final section, which displays after all TCBs are listed, is a summary for the Private Area:

Tot-Hi
Summary of LSQA allocated.

Tot-Lo
Summary of user virtual storage allocated.

Totals
Summary of LSQA and user virtual storage allocations.

This example shows that the program IFOX00 currently has 28K bytes of storage that is allocated to subpool 251, key 8. It is allocated in three blocks and the storage that is defined by the last block begins at DD000. Of all subpool 251 storage, EF0 bytes are free (not in use). The EF0 free bytes are made up of three blocks. The largest of these blocks is 968 (hex) bytes long.

TCBS
Displays the current TCB structure for the target user.

Type:
Minor of the PEEK command

This next screen shows a typical TCBS display.

```
 PEEK  PAYROLL  ASID=25, collected at 15:39:39
==============================================================================
 tcbs   Program  Mother   Daughter  Sister  Jobstep
+       IEAVAR00          IEESB605          (SELF) (Region Control Task)
+       IEESB605 IEAVAR00 IEFIIC    IEAVTSDT (SELF) (Started Task Control)
+       IEFIIC   IEESB605 PAY1              (SELF) (Initiator)
+       PAY1     IEFIIC                     (SELF)
+       IEAVTSDT IEAVAR00                   (SELF) (SVC Dump Task)
==============================================================================
```

Figure 193. TCBS display

A typical TCB structure for a batch job follows:

Program
Load module name of the most recently created RB for each TCB. In this case, PAY1 indicates the name on the EXEC PGM= parameter.

Mother Daughter Sister
Program names for the mother, daughter, and oldest sister TCBs of the Program TCB.

Most address spaces on this level point to themselves as the Jobstep TCB. You can use this information to easily construct a picture of the current TCB structure as shown in the following example:

```
         +----------+
         |          |
         | IEAVAR00 |
         |          |
         +----------+
              |
         +----------+         +----------+
         |          |         |          |
         | IEESB605 |---------| IEAVTSDT |
         |          |         |          |
         +----------+         +----------+
              |
         +----------+
         |          |
         |  IEFIIC  |
         |          |
         +----------+
              |
         +----------+
         |          |
         |   PAY1   |
         |          |
         +----------+
```

Figure 194. Example of TCB structure

If you place an A in the label field of TCBS, OMEGAMON for IMS displays the actual TCB address under each TCB program name. This information might be useful if several TCBs in the same step start the same program.

If you place an X in the label field of TCBS, two extra lines display for each TCB. The first line shows the storage protect key for the TCB and indicates whether the address space is APF-authorized. The second line indicates either DISPATCHABLE or NON-DISPATCHABLE. For those TCBs that cannot be dispatched, a short explanation displays to indicate which non-dispatchability bit is found set.

For example, the following message indicates that the TCB is merely waiting to post an ECB.

```
        NON-DISPATCHABLE: TOP RB WAITING ON ECB
```

This is the most common reason for a TCB to be non-dispatchable.

You can optionally specify a two-digit number in the operand field of TCBS to suppress the display of the first *nn* TCBs. This number is useful if all of the TCBs do not fit on one screen.

Important: If the job that is PEEKed has many TCBs, you might must increase the maximum number of control blocks that are tested with the LOOPCOUNT keyword of the **.SET** command.

Online facility for logging IMS messages

You can use the facility in OMEGAMON for IMS that copies IMS MTO messages and IMS-related MCS (multiple console support) messages into SYSOUT files. This facility is known as the OMEGAMON for IMS message logging facility.

By using the message logging facility, you can browse current IMS messages with the IBM SDSF® or similar products, and print IMS messages. You can also create permanent historical copies of IMS messages by using the external writer facilities of JES.

The message logging facility does not require any additional DD statements in the OMEGAMON for IMS procedure. All SYSOUT message log files are dynamically allocated. The SYSOUT class to be used for the message log files are defined during message logging facility startup.

Installation of the message logging facility

Although the message log file does not require any additional installation steps, planning is needed to make effective use of the facility. You must select a JES SYSOUT class to be used by the message logging facility. Then you can create external writer procedures to capture the SYSOUT files that the message logging facility writes to that class.

However, you must ensure that the selected SYSOUT class must be used exclusively by the message logging facility. This is necessary to avoid collision between message logging facility output and other miscellaneous system output during the external writing process.

You must also determine the destination of output that is captured by the external writer procedure. This output might go to tape, generation data groups, sequential disk data sets, or microfiche. This destination must be specified in the external writer task that is used to capture output from the message logging facility.

You can start the external writer task at any time. It is advisable to start when output from the message logging facility should be removed from the JES spool.

A sample external writer procedure is shown in Figure 195 on page 303.

```
//MLOGXWTR PROC
//IEFPROC EXEC PGM=IASXWR00,REGION=64K,
//             PARM='PW'   <<<<<<------------ CLASS W IS FOR MLOG
//IEFRDER  DD  DSN=CANDLE.MLOG,DISP=(MOD,PASS),
//         DCB=(BLKSIZE=13030,LRECL=137,BUFL=13030,BUFNO=2,RECFM=VBM),
//         UNIT=TAPE,VOL=SER=TVOL01
```

Figure 195. Typical JCL for external writer procedure

To start an external writer task, issue the following (or similar) command at an z/OS console:

```
        START MLOGXWTR.MLOGWTR
```

To stop the external writer task, issue the following (or similar) command at an z/OS console:

```
STOP MLOGWTR
```

For more information about the external writer facility, see the *OS/VS2 z/OS System Programming Library: Job Management*.

Modifying the message logging facility

Issue the `MLOG` authorized command to examine or alter the online facility for logging IMS messages.

MLOG
Displays or alters status of facility for logging IMS messages.

Type:
Immediate

Format:
-MLOG opr

The operand (*opr*) is one of the following operands:

FLUSH=n
If specified, this operand sets up the automatic FLUSH threshold. When the message logging facility reaches this threshold, it automatically closes the SYSOUT data set, make it available to the external writer task, and reallocate it.

n: might be a number between 0 - 65,535 inclusive. When *n* is set to zero, all flush processing is canceled immediately. When *n* is set to a number between 1 - 65,535, flush processing occurs after the message logging facility logs *n* messages.

The =n portion of the operand is optional.

If =n is not specified, the FLUSH operand causes the current SYSOUT message log file to be closed and reallocated immediately by the message logging facility. This makes the message log file available to be processed by the external writer task.

START
The START operand causes the message logging facility to start.

The format is:

```
-MLOG START, CLASS=c
```

The optional CLASS suboperand specifies the default SYSOUT class (*c*) for the message log. If you do not specify an output class with the CLASS suboperand, it defaults to your installation's default output class.

STOP
The STOP operand causes the message logging facility to initiate shutdown processing. The current message log file is closed and is made available to the external writer task. The message logging facility is then rendered inactive.

STATUS
By default, the STATUS operand displays the status of the message logging facility.

- The status that is provided includes the following information:
- The status of the message logging facility:
  - active
  - inactive
  - disabled
  - pending critical operations
- The SYSOUT class.
- The number of records that are written to the current SYSOUT data set by the message logging facility since start up or since the last message log file flush.
- The automatic flush threshold, if automatic flushing is in effect.

The next figure shows a typical example of an MLOG command.

```
-MLOG START, CLASS=D
```

Dynamically allocating and deallocating data sets

You can use the DYNA command to dynamically allocate a data set and the DYNU command to dynamically deallocate a data set.

DYNA
Dynamically allocates a data set.

Type:
Immediate

You must run the DYNA command twice to allocate a data set:

1. Specify the data set name.
2. Allocate the data set.

The format of the DYNA command is:

```
        aDYNA c...c
```

a
Specifies the action of the command:

b
Defines the data set to be allocated. If this field is blank, c...c must specify the data set allocation parameters.

-

Allocates the data set. If this field contains the action character (-), c...c must be blank.

c...c

Specify the data set to be allocated by using the following parameters:

DD=ccccccc

Specifies the ddname.

UNIT=c

Specifies the unit name.

An asterisk (*) allocates a cataloged data set.

VOLSER=cccccc

Specifies the volume serial number. To allocate a cataloged data set, enter an asterisk (*).

DISP=

Specifies the initial disposition of the data set (OLD, SHR, or MOD).

DSN=cccccccc

Specifies the data set name.

If this field contains the data set description, the label field (*a*) must be blank.

If this field is blank (if the data set name is already specified), you must specify the action character (-) in the label field to allocate the data set

In the next figure, the DYNA command specifies a cataloged data set, with a ddname of MYDD, an initial disposition of SHR, and a data set name of MY.DATASET.

```
DYNA DD=MYDD,UNIT=*,VOLSER=*,DISP=SHR,DSN=MY.DATASET
```

Figure 196. Dynamically allocating a data set

Enter DYNA again to allocate the data set specified previously.

```
        -DYNA
```

DYNU

Dynamically deallocates a data set.

Type

Immediate

You must run the DYNU command twice to deallocate a data set:

1. Specify the data set name.
2. Deallocate the data set.

The format of the DYNU command is as follows:

```
        aDYNU c...c
```

a

Specifies the action of the command.

b

Defines the data set to be deallocated. If this field is blank, c...c must specify the data set name.

-

Deallocates the data set. If this field contains the action character (-), c...c must be blank.

c...c

May specify the data set to be deallocated:

NAME=ddname
Specifies the data set name. If this field contains the data set name, the label field (.a) must be blank.

b
If this field is blank, that is, if the data set name is already specified, you must specify the action character (-) in the label field (X'a') to deallocate the data set.

In the following example, the DYNU command selects the data set MYDD.

```
        DYNU DD=MYDD
```

The DYNU command in the next example deallocates the data set specified previously.

```
        -DYNU
```

Displaying and modifying data space and hiperspace storage

You can use OMEGAMON for IMS commands to display or modify IMS data space and hiperspace storage for systems that use the capabilities of z/OS and DFP 3.1 to create these data-only spaces.

The following commands parallel the functions of the **MLST**, **MCHN**, **MSCN**, and **MZAP** storage commands. Because of the potential security risk that is associated with using these commands, the **.DSA** command exists to provide an extra level of protection.

.DSA
Sets and displays authorization to list and zap non-shareable data-only spaces.

Type:
Immediate

The **.DSA** command provides a mechanism to limit the scope of the listing and zapping commands to shareable data-only spaces (data spaces or hiperspaces that are defined by the owner as able to be shared by other address spaces).

Command operands:

ON
Turns on data-only space authorization, that is, access is allowed to all data-only spaces.

OFF
Turns off data-only space authorization, that is, access is restricted to shareable spaces only.

Entering the **.DSA** command with no operand displays the status of data-only space authorization.

OSPC
Lists the attributes of the owner of a data-only space.

Type:
Immediate

Format:
OSPC spacename

Where *spacename* is the name of the data-only space you want to list. If you do not enter a name, OSPC lists all data spaces and hiperspaces. You can also enter any number of characters from 1 to 7, and OSPC displays any space names that begin with the character string entered.

OSPC provides the following information about the specified data space or hiperspace:

- Type of data-only space
- ASID of owning TCB
- Jobname of owning TCB
- Address of owning TCB

Here is an example of output from the **OSPC** command.

```
 >OSPC MYSPACE
+  TYPE        ASID     JOBNAME        TCB address   SPACE NAME
+ Data space     12      MYJOB           007FFA10      MYSPACE
+ Hiperspace     22     HISJOB           007B7CB0      MYSPACE
```

Figure 197. OSPC command display

SLST

Displays bytes of memory from data-only space storage.

Type:
Immediate

Format:
aSLSTc jobname,spacename,addr,plen

a
A required action character in column 1:

-
Changes to a comment character (>) after command runs.

<
Does not change to a comment character after command runs. Use this action character to repeat the command.

c
Specifies the format of the output:

B or blank
Memory dump format (default)

C
Character only

X
Hex only

jobname
The jobname or ASID in decimal of the owner of the data-only space.

spacename
The name of the data-only space.

addr
The starting address of the data.

plen
The number (1 - 8 hex digits) of bytes to print. The default is 16 (X'10') bytes or 1 line.

SLST lists memory from data-only spaces. When necessary, an SRB is scheduled into the address space of the TCB owning the data-only space to be listed.

Here is an example of using SLST.

```
<SLST MYJOB,MYSPACE,1000,20
+Storage at 00001000 in dataspace MYSPACE, job MYJOB ASID=12
+ 0000  E3C5E2E3 40C4C1E3 C140E2D7 C1C3C540  *TEST DATA SPACE *
+ 0010  F0F1F2F3 F4F5F6F7 F8F9C1C2 C3C4C5C6  *0123456789ABCDEF*
```

Figure 198. SLST command display

SSCN

Scans data-only space storage for the occurrence of a specific string of data.

Type:
Immediate

Format:
aSSCNc jobname,spacename,addr,string,len1,len2

a
A required action character in column 1:

-
Changes to a comment character (>) after command runs.

<
Does not change to a comment character after command runs. Use this action character to repeat the command.

c
Specifies the format of the output:

B or blank
Memory dump format (default)

C
Character only

X
Hex only

jobname
The jobname or ASID in decimal of the owner of the data-only space.

spacename
The name of the data-only space.

addr
The starting address of the scan. See “Address specification for storage commands” on page 272 for more options on specifying addr.

string
The comparison string for the scan. Either a hexadecimal string or a character string. Use single quotation marks around a character string. Do not use quotation marks around a hexadecimal string.

len1
The length to scan in hex. Default is 256 (X'100') bytes.

len2
The length of print display. Default is one line or 16 (X'10') bytes.

SSCN scans data-only space storage until a match to the string is found or the length of storage that is specified is exhausted. When necessary, an SRB is scheduled into the address space of the TCB owning the data-only space to be scanned.

The following figure is a sample SSCN display.

```
>SSCN MYJOB,MYHIPER,1000,’TEST’,200,20
+Storage at 00001100 in hiperspace MYHIPER, job MYJOB ASID=12
+ 0000  E3C5E2E3 40C8C9D7 C5D9E2D7 C1C3C540  *TEST HIPERSPACE *
+ 0010  F0F1F2F3 F4F5F6F7 F8F9C1C2 C3C4C5C6  *0123456789ABCDEF*
```

Figure 199. SSCN display

SCHN
Scans data-only space control blocks for a string of data and displays the location.

Type:
Immediate

This command is used to search chained control blocks that in a data-only space for the occurrence of a specific string of data.

Format:
aSCHNc jobname,spacename,addr,string,off1,off2,plen

a

A required action character in column 1:

-

Changes to a comment character (>) after command runs.

<

Does not change to a comment character after command runs. Use this action character to repeat the command.

c

Specifies the format of the output:

B or blank
Memory dump format (default)

C

Character only

X

Hex only

jobname
The jobname or ASID in decimal of the owner of the data-only space.

spacename
The name of the data-only space.

addr
The starting address of the scan. See "Address specification for storage commands" on page 272 for more options on specifying addr.

string
The comparison string for the scan. Either a hexadecimal string or a character string that is surrounded by single quotation marks.

off1
The offset from the beginning of the control block to the location of the comparison string. This value might be preceded by a + or - sign.

off2
The offset from the beginning of the control block to the fullword address of the next control block. This value might be preceded with a + or - sign.

plen
The length of print display. Default is one line or 16 (X'10') bytes.

SCHN scans data-only space storage until either a match to the string is found, the chain loops, or the address of the next control block is zero. When necessary, an SRB is scheduled into the address space of the TCB owning the data-only space to be scanned.

The following example shows the SCHN display for a data space.

```
>SCHN MYJOB,MYSPACE,1000,'TEST',0,30,20
+Storage at 00001100 in dataspace MYSPACE, job MYJOB ASID=12
+ 0000  E3C5E2E3 40C4C1E3 C140E2D7 C1C3C540  *TEST DATA SPACE *
+ 0010  F0F1F2F3 F4F5F6F7 F8F9C1C2 C3C4C5C6  *0123456789ABCDEF*
```

Figure 200. SCHN command display

SZAP
Modifies the contents of data-only space storage.

CAUTION: There is a potential integrity exposure when SZAP on hiperspaces is used. SZAP uses HSPSERV to read in a page of data from the target hiperspace, to check the data, to alter the data, and to use HSPSERV to write the page back to the hiperspace. If someone else is writing to the same page of the hiperspace while this process is occurring, the newly entered data might be lost. There is no available enqueue mechanism to guard against this exposure.

Type:
Immediate

Format:
-SZAP jobname,spacename,addr,vercode,repcode

-
Action character that is required for execution.

jobname
The jobname or ASID in decimal of the owner of the data-only space.

spacename
The name of the data-only space.

addr
The starting address of the data. See “Address specification for storage commands” on page 272 for more options on specifying addr.

vercode
The current code to be verified in hexadecimal.

repcode
The replacement code in hexadecimal.

The lengths of vercode and repcode must match.

When necessary, an SRB is scheduled into the address space of the TCB owning the data-only space to be zapped. The following example shows typical SZAP command output.

```
>SZAP MYJOB,MYSPACE,1000,00000005,00000008
>      >> OB7110: Memory Zap Successful <<
```

Figure 201. SZAP

Chapter 13. OTMA information

Use the OTMA, TMEM, and TPIP command to monitor OTMA status.

OTMA
: Displays general information about the state of OTMA.

Type:
Immediate

Format:
OTMA

The OTMA command has no parameters. General status information displays as follows:

- XCF group name
- IMS TMEMBER name
- Number of messages enqueued
- Security option
- Count of asynchronous and synchronous conversations
- OTMA status and XCF status
- Status of the server
- Shutdown status if OTMA is shutting down
- Input and output status
- Input and output trace status
- Input queue count and rate (IMS V10 and higher)
- Output queue count and rate
- Count of TMEMBERS
- Count of TPIPEs

The following figure shows an example of OTMA status information.

```
 OTMA
+ ==============================================================================
+  XCF Group Name.....: I91CG             IMS TMEMBER Name..: IMS910C
+  Messages Enqueued..: 0                 Security Option...: None
+  Async Conversations: 0                 Sync Conversations: 0
+  OTMA Status........: Enabled           XCF Status........: Active
+  Server Status......: Active            Shutdown Status...: N/A
+  Input Status.......: Normal            Output Status.....: Normal
+  Input Trace........: Inactive          Output Trace......: Inactive
+  Input Queue Count..: N/A (V10)         Output Queue Count: 0
+  Input Queue Rate...: N/A (V10)         Output Queue Rate : .00
+  TMEMBER Count......: 2                 TPIPE Count.......: 12
+ ============================================================================
```

Figure 202. OTMA status display

TMEM
: Displays information about OTMA TMEMBERs.

Type:
Immediate

Format:
TMEM [MEM=*tmembername*] [XSTATUS=*xcfstatus*] [USTATUS=*userstatus*] [SMEM=*smembername*]

The TMEM command with no parameters displays information about all TMEMBERs. The TMEM display can be filtered by specifying one or more of the following optional parameters:

MEM=*tmembername*

Specifies the name of the TMEMBER for which information is to be displayed. Maximum length of *tmembername* is 16 characters.

XSTATUS=*xcfstatus*

Specifies the XCF status of a TMEMBER; only TMEMBERs having an XCF status that matches the value that is specified by *xcfstatus* is displayed. The value of *xcfstatus* can be one of the following values:

- ACTIVE
- CREATED
- FAILED
- NOTDEFINED
- QUIESCED

USTATUS=*userstatus*

Specifies the user status of a TMEMBER; only TMEMBERs having a user status that matches the value that is specified by *userstatus* is displayed. The value of *userstatus* can be one of the following values:

- ACCEPTTRAFFIC
- DISCONNECTED
- FLOOD
- INSLOWDOWN
- SERVER
- SMQBACKEND
- STOPINPUT
- SUPERMEMBER
- WAITBID
- WAITRESPONSE

SMEM=*smembername*

Specifies the super member name that is used by a TMEMBER; only TMEMBERs by using the super member name that matches the value that is specified by *smembername* is displayed. The maximum value of *smembername* is 4 characters.

Figure 203 on page 313 shows an example of issuing the TMEM command (without parameters) to display all TMEMBERs.

Note: The TIB Count, Maximum TIBs, TIB Used%, TIB Rate, Timeout CM1, and DRU Exit do not display for a TMEMBER that is a server or super member as these values are not applicable.

```
 TMEM
+ ============================================================================
+  TMEMBER Name: IMS910C          XCF Status..: Active      Security : None
+  User Status : Server           Super Member:             Trace....: Off
+  Input Queue Count: N/A (V10)   Output Queue Count: 0
+  Input Queue Rate : N/A (V10)   Output Queue Rate : .00
+  TPIPE Count......: 0
+ ============================================================================
+  TMEMBER Name: I91C             XCF Status..: Active      Security : None
+  User Status : Accept Traffic   Super Member:             Trace....: Off
+  TIB Count........: 0           Maximum TIBs: 0           TIB Used%: .00
+  TIB Rate.........: .00         Timeout CM1 : N/A (V10)   DRU Exit : HWSYDRU0
+  Input Queue Count: N/A (V10)   Output Queue Count: 0
+  Input Queue Rate : N/A (V10)   Output Queue Rate : .00
+  TPIPE Count......: 12
+ ============================================================================
```

Figure 203. OTMA TMEMBER status display

TPIP

Displays information about OTMA TPIPEs.

Type:

Immediate

Format:

TPIP MEM=*tmembername* [PIPE=*tpipename*] [STATUS=*status*] [QCOUNT=*nnnnnn*]

MEM=*tmembername*

Required parameter. Specifies the name of the TMEMBER for which TPIPE information is to be displayed. Maximum length of *tmembername* is 16 characters.

Information for all TPIPEs for the specified TMEMBER is displayed. The following optional parameters are provided to allow the TPIPEs for a TMEMBER to be filtered:

PIPE=*tpipename*

Specifies the name of the TPIPE for which information is to be displayed. Maximum length of *tpipename* is 8 characters.

STATUS=*status*

Specifies the status of a TPIPE; only TPIPEs having a status that matches the value that is specified by status is displayed. The value of *status* can be one of the following values:

- DEQFAIL
- FLOOD
- RESETFAIL
- SENDREQR
- STOPINPUT
- STOPPED
- SYNC
- TEMP
- WAITA
- WAITH
- WAITREPR
- WAITTBR

QCOUNT=*nnnnnn*

Specifies the output queue count for a TPIPE; only TPIPEs having an output queue count greater than or equal to *nnnnnn* is displayed.

Figure 204 on page 314 displays the TPIPEs for TMEMBER I91C.

```
 TPIP MEM=I91C
+ ==============================================================================
+  TPIPE Name...: TEST0000                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: Yes  StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: TEST0002                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTILL04                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: Yes  StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No


Press PF8 to scroll forward.
```

Figure 204. OTMA TPIPE status display (panel one of four)

Figure 205 on page 314 displays more OTMA TPIPE status information.

```
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTILL03                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTILL05                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTILL02                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: Yes  StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No


Press PF8 again to scroll forward.
```

Figure 205. OTMA TPIPE status display (panel two of four)

Figure 206 on page 315 displays more OTMA TPIPE status information.

```
+ ==============================================================================
+  TPIPE Name...: TEST0004                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTILL01                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: TEST0003                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: Yes  StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================


And PF8 to scroll forward again - this is the last page.
```

Figure 206. OTMA TPIPE status display (panel three of four)

Figure 207 on page 315 displays the final panel of OTMA TPIPE status information.

```
+  TPIPE Name...: RSAPPL01                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: RSTIL00                            Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
+  TPIPE Name...: TEST0001                           Input  Que Cnt : N/A (V10)
+  Enqueue Count: 0         Dequeue Count: 0         Output Que Cnt : 0
+  Async Enq Cnt: 0         Async Deq Cnt: 0         Async  Que Cnt : 0
+  Sync  Enq Cnt: 0         Sync  Deq Cnt: 0         Sync   Que Cnt : 0
+  TPIPE Trace..: Off                                Input  Que Rate: N/A (V10)
+  Enqueue Rate : .00       Dequeue Rate : .00       Output Que Rate: .00
+                           T P I P E   S T A T U S
+  Stopped: No   StopInput: No   Flood: No   WaitA: No   WaitH: No   Sync: No
+  DeqFail: No   ResetFail: No   Temp : No   SREQR: No   WREPR: No   WTBR: No
+ ==============================================================================
```

Figure 207. OTMA TPIPE status display (panel four of four)

Chapter 14. OMEGAMON for IMS automation and logging features

Before you try to use the event- and time-driven features, be familiar with how exception analysis works and how to create screen spaces. The following topics describe these features and the commands that control them.

Event- and time-driven features

OMEGAMON for IMS can automatically start displays, initiate action, and log information in response to exception conditions or at specific times. When an automated sequence ends, OMEGAMON for IMS resumes normal operation without manual intervention.

The three event- and time-driven features are as follows:

XLF (exception logging facility)
This feature automatically time-stamps and logs exception messages for your review. You can correct intermittent performance problems by documenting the frequency and severity of system-wide exceptions.

ASF (automatic screen facility)
This feature automatically starts a predefined screen space when an exception occurs for more than a specified number of successive cycles. The predefined screen space can contain commands to turn on the log, further evaluate the exception condition, and complete other options.

TSF (timed screen facility)
This feature automatically starts screen spaces at times or time intervals you specify. Many sites use TSF to spin off copies of the REPORT and of XLFLOG files to the printer. In general, you can use the TSF facility to automate many day-to-day housekeeping routines.

These three event- and time-driven features require OMEGAMON for IMS to be running in dedicated or VTAM automatic update mode.

Controlling automating features

You can use profile options and OMEGAMON for IMS commands to control the XLF, ASF, and TSF features.

If you are operating OMEGAMON for IMS with the menu interface, choose the **PROFILE** option from the main menu. Then, you can choose the following options:

- Options listed under **Exceptions**, to define XLF and ASF parameters for each individual exception.
- **BACKGROUND** to control the operation of the three background processing features, and define entries for TSF.
- **LOGGING** to turn the log on and off, and to spin the log out to the printer.

The following commands control the XLF, ASF, and TSF features.

XLF commands

Use the `XACB`, `OPTN`, and `/XLFOUT` commands to set up and operate the XLF feature.

XACB
Sets parameters and controls the operation of XLF for each individual exception. The XLF keywords are explained in "Setting parameters for XLF and ASF operation" on page 318.

OPTN
Turns the XLF feature ON and OFF (with the XLF keyword).

/XLFOUT
Sends the XLF data to the JES output queue and reallocates the output file. If you want to save the command in a screen space, use the **.XLFOUT** immediate command instead of the **/XLFOUT** INFO-line command.

ASF commands

Use the XACB, OPTN, and /LOGOUT commands to set up and operate the ASF feature.

XACB
Sets parameters and controls the operation of ASF for each individual exception. The ASF keywords are the same as the XLF keywords and are explained in "Setting parameters for XLF and ASF operation" on page 318

OPTN
Turns the ASF feature ON and OFF (with the ASF keyword).

/LOGOUT
Sends the REPORT data to the JES output queue and reallocates the output file. If you want to save the command in a screen space, use the **.LOGOUT** immediate command instead of the **/LOGOUT** INFO-line command.

TSF commands

Use the .TSF, OPTN, and /LOGOUT commands to set up and operate the TSF feature.

.TSF
Defines entry for TSF. This command is described in "Setting parameters for TSF operation" on page 319.

OPTN
Turns the TSF feature ON and OFF (with the TSF keyword).

/LOGOUT
Sends the REPORT data to the JES output queue and reallocates the output file. If you want to save the command in a screen space, use the **.LOGOUT** immediate command instead of the **/LOGOUT** INFO-line command.

Setting parameters for XLF and ASF operation

Use the **XACB** command to set the parameters for XLF and ASF operation and to activate the XLF and ASF features for all occurrences of any exception, or only if the exception persists for a specified number of cycles. You can also set a limit on the number of times an exception starts the XLF or ASF feature.

The following figure shows the format of the **XACB** command. The variable *cccc* is the exception name. To dynamically set parameters for an exception, type over the current value that displays when the **XACB** command is issued.

```
 XACB LIST=cccc
: cccc
+  DISPLAY Parameters:      THRESHOLD Parameters:     XLF Parameters:
:    State=                  Threshold=                Auto=
:    Group=                  Display=                  Log=
:    Bell=                   Attribute=                Limit=nnnnn
+  BOX Parameters:          CYCLE Parameters:          Repeat=
:    Boxchar=' '             ExNcyc=n                  Persist=nn
:    Boxclr=                 Stop=n (m)                Sc=
:    Boxattr=                Cumulative=n
```

Figure 208. XACB command display

The XLF and ASF parameters are as follows:

AUTO
Controls the status of ASF for this exception (ON/OFF).

LOG
Controls the status of XLF for this exception (ON/OFF). It does not affect logging for ASF.

LIMIT
Limits the number of times *(nnnnn)* XLF and ASF starts if the exception occurs. If you specify `Limit=00`, no events are logged. If you specify `Limit=NONE`, XLF and ASF starts each time that the exception occurs. The maximum setting for limit is 32726. The parenthetical number to the right of this parameter indicates the remaining number of times that the exception is logged. You can reset the LIMIT parameter to continue logging the exception.

REPEAT
Used with the PERSIST threshold, `Repeat=YES` specifies that XLF logging or ASF action occurs each time the PERSIST threshold is reached. For example, if `Persist=5` the exception condition persists for 15 cycles, and logging is in effect, then the message would be logged three times. If **Repeat=NO**, the message is logged once only as specified with the PERSIST parameter.

PERSIST
Logs the exception message and starts the ASF screen spaces when the condition persists for *nn* consecutive OMEGAMON for IMS cycles. After it reaches the threshold, the message is logged once only (or ASF starts once only), unless the condition stops for at least one cycle and then trips again. If you specify **Persist=00**, no events are logged. The default is 0 cycles.

Sc
You can dynamically alter the second character of this keyword to either an S or an L. You must specify this parameter for ASF to work.

SL
Specifies the screen space to start if ASF is in effect **(Auto=ON)**, and specifies that the output of the ASF screen spaces is to be logged. The OMEGAMON for IMS REPORT log automatically turns on when the exception trips, and screen space logging starts.

SS
Specifies the screen space to start if ASF is in effect. It does not turn on the log.

Setting parameters for TSF operation

The Timed Screen Facility (TSF) schedules screen spaces to run at specified times of day or at specified intervals. You define the screen spaces, and then set the times when you want them to run.

For example, you can turn on bottleneck analysis automatically at 2:00 PM every weekday, and then turn it off again at 2:15. You can also start a screen space at regular intervals, such as every hour.

Follow this procedure to set up and operate the TSF feature.

1. Create any screen spaces that you want TSF to start. You can use the `.SGO` or `.FGO` command to chain screens together, and the `.RTN` command to end the cycle.
2. Use the **.TSF** command to enter the names of the screen spaces and the times or time intervals when you want them to run.

3. Turn on TSF with the TSF keyword of the **OPTN** command.
4. Set your terminal in automatic update mode by using the **.AUPON** command.
5. Do not enter any more commands.

CAUTION: If you enter commands, a TSF cycle might be interrupted and screen spaces that must run on that cycle do not run.

Changing TSF definitions

Use the **.TSF** command to view and change TSF definitions. A TSF definition includes the name of a screen space that you want to start and when you want it to run.

1. Use **.TSF00** to display the names of screen spaces and the time or intervals when they run. The TSF table is shipped with 99 blank entries as the following example shows.

```
.TSF00
        1  Time=0000  SS=*NONE*  DAY=DAILY
        2  Time=0000  SS=*NONE*  DAY=DAILY
        3  Time=0000  SS=*NONE*  DAY=DAILY
        4  Time=0000  SS=*NONE*  DAY=DAILY
```

Figure 209. TSF table

Use **.TSF00nn** to skip the first *nn* entries, and display the remaining entries. For example, to display entries 21 through 99 in the TSF table, skipping the first 20 entries, enter **.TSF0020**.

2. To set the definitions for a specific entry, enter **.TSFnn** to display entry *nn* in the TSF table. For example, type **.TSF01** and press **Enter** to display the first entry in the table.

```
.TSF01  Time=0000  SS=*NONE*  DAY=DAILY
```

Type over the settings, and then press **Enter** to change the definitions for the entry.

3. Alternatively, you can change the definitions for entry *nn* by directly typing the Time, SS or SL, and DAY keywords and their values on the **.TSFnn** command.

The following keywords and their values can be specified on the **.TSFnn** command.

Table 13. .TSFnn keywords

.TSFnn keyword	Description
Time	Specifies the time of day (0000 - 2400) to start the screen space. TIME=+*nn* starts the screen space every *nn* minutes. **Note:** The screen space does not run while the TIME=+nn entry remains on your current screen.
SL or SS	Specifies the screen space to start if TSF is in effect. SS specifies the screen space to start, but does not turn on the log. SL automatically turns on the REPORT log when the exception trips, and screen space logging starts. You must specify either SS or SL for TSF to work.
DAY	The valid entries for day of week are MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY, WEEKDAY, WEEKEND, and DAILY. The days of the month are also valid entries (numerals 1 - 31). The default value is DAILY. You can abbreviate the input if it is unique, and if the day of the week is recognized. You can specify day combinations by enclosing the names of the days within parentheses, and by separating each day with either a comma or a blank.

The following example shows five TSF entries.

```
.TSF01   1   TIME=1800 SL=WENDSHFT  DAY=DAILY
.TSF02   2   TIME=0600 SS=WEEKSTRT  DAY=MONDAY
.TSF03   3   TIME=1200 SL=NOONTIME  DAY=(TH,F)
.TSF04   4   TIME=0800 SS=MONTHEND  DAY=30
.TSF05   5   TIME=+30  SL=STATUS    DAY=DAILY
```

Figure 210. TSF entries

These entries perform the following functions.

TSF01
Runs and logs screen space WENDSHFT, at 6:00 PM daily.

TSF02
Runs screen space WEEKSTRT, at 6:00 AM every MONDAY.

TSF03
Runs and logs screen space NOONTIME, at 12:00 PM every Thursday and Friday.

TSF04
Runs screen space MONTHEND, on the 30th of each month at 8:00 AM.

TSF05
Runs and logs screen space STATUS, every 30 minutes, every day.

Log parameters for XLF, ASF, and TSF

Exception messages that occur when XLF is running are routed to the XLFLOG file. Screen space output from ASF and TSF is routed to the REPORT file. You can display or change the logging parameters for XLF, ASF, and TSF with the OUTP major command and its minor commands.

For XLF, specify OUTP XLFLOG. For ASF or TSF, specify OUTP REPORT.

In XLF and ASF processing, only one record is written to the log while an exception condition persists. However, new records are written to the XLFLOG or the REPORT log if any of the following situations occur:

- If an exception does not display for even one cycle, and then displays again, it is considered a new event.
- If you turn off an exception and then on again, you clear the event. If the exception condition still exists, another record goes to the XLF log. OMEGAMON for IMS does not check for the condition unless the exception is turned on.
- If you turn off XLF or ASF and then turn it on again, you clear all events and new records go to the log.
- For an exception, if you set the XACB REPEAT parameter to YES and specify a threshold for the PERSIST parameter, a new record is written each time the PERSIST threshold is reached.

Screen spaces in ASF and TSF

The power of the ASF and TSF features is the capability to branch to, run, and log an analysis screen or series of screens. The screen space that you specify to be called when ASF is activated can contain whatever information-gathering and action-taking commands that you specify. You can include commands to turn on the log, change OMEGAMON for IMS defaults, further analyze the exception condition, or even call other screen spaces (with the `.SGO` or `.FGO` commands).

How ASF and TSF screen spaces run

If two exceptions occur at the same time, screen spaces and logging run for the first exception until the final screen space issues the `.RTN` command. If the second exception persists, the automatic mode reactivates on the next cycle.

If XIMS displays on any of your target screen spaces, any triggered exception messages display as usual, but ASF ignores these new exceptions until the sequence is ended with the **`.RTN`** command. If the exception condition still exists, ASF proceeds to the next exception in sequence.

If you want to cancel an ASF or TSF sequence while it is running, press any key other than a cursor key. If, for example, you press Enter, the sequence immediately ends. However, when it ends without a **.RTN** command, OMEGAMON for IMS does not return to the original calling screen, or to another screen defined with .RTN.

You can start either named or numbered screen spaces with ASF and TSF. By assigning a screen to a PF key in virtual storage, you assure availability even when the I/O subsystem is not functioning because of a problem.

The TSF sequences always function at their scheduled times, except when an ASF sequence is also tripped. ASF sequences take precedence over TSF. That means that if an ASF sequence trips while a TSF sequence is in progress, the ASF sequence cancels the TSF sequence. When the ASF sequence is complete, OMEGAMON for IMS returns to the screen displayed at the time that ASF tripped, but the TSF sequence does not continue. Similarly, if a TSF sequence is scheduled while an ASF sequence is already in progress, the TSF sequence is ignored.

The .RTN command

You can branch to as many screen spaces as you want in ASF and TSF by using the .FGO or .SGO command to chain screens together. The **.RTN** immediate command is required at the end of the last screen space in an ASF or TSF sequence to end the sequence, return to the original calling screen or branch to the next, and re-enable exception analysis for further automatic calls. You should use definition mode (/DEF ON) when creating screen spaces that contain the .RTN command.

While the **.RTN** command is normally used to return to the calling screen space, it also accepts an argument, and can force the return to a screen space that is not the calling screen. For example, to return to a screen called **SCREEN2**, enter the .RTN immediate command followed by **SCREEN2**.

You can delay the return for up to 35 cycles by placing the number of cycles to be delayed in the label field of the **.RTN** command (1-9 for numbers 1-9 and A-Z for numbers 10-35). The following command causes a return to **SCREEN2** after 6 cycles:

```
6.RTN SCREEN2
```

On each cycle, OMEGAMON for IMS replaces the number in the label field with the next lesser number. When the count reaches zero, OMEGAMON for IMS fetches **SCREEN2**.

ASF example

You must set the ASF keyword of the OPTN command to ON to enable the automatic screen feature. In addition, you must be running OMEGAMON for IMS in dedicated mode or VTAM mode with automatic updates in effect.

The following example shows some parameters that are set with the XACB command for the DRDY exception.

```
XLF Parameters:
 Auto=ON
 Log=OFF
 Limit=3
 Repeat=NO
 Persist=5
 SL=DEX01
```

Figure 211. XLF parameters

In this example, when the exception condition exists for 5 cycles in a row, ASF starts screen space DEX01. From this point on, an A displays in the far right portion of the INFO-line, which indicates that the current screen is part of an ASF sequence.

The following example shows the DEX01 sequence of commands:

```
 _______________ DEX01    VTM     OI-II    V530./C I91C 04/09/15 14:03:33    B
> After 8 cycles, branch to screen space DEX02
8.SGO DEX02
===============================================================================
> Reset the OMEGAMON for IMS cycle time to 15 seconds
 .SET INTERVAL=15
===============================================================================
> Start degradation analysis
 IDEG
 begn
===============================================================================
> Display degradation analysis for performance group 2
 pdex02
```

Figure 212. DEX01 command sequence

In this example, the OMEGAMON for IMS cycle time is set to 15 seconds. Because the 8.SGO entry waits 8 cycles before it jumps to DEX02, bottleneck analysis runs on this screen for 2 minutes and logs the results. Then, screen space DEX02, shown in Figure 213 on page 323, starts.

```
 _______________ DEX02    VTM     OI-II    V530./C I91C 04/09/15 14:13:16    B
===============================================================================
> Suspend degradation analysis
 IDEG
 susp
===============================================================================
> Return to the calling screen
 .RTN
```

Figure 213. DEX02 screen space

The DEX02 screen space suspends degradation analysis, and returns to the calling screen with the `.RTN` command. Upon return, the ASF sequence ends (the **A** no longer displays from the INFO-line), enabling XIMS for further automatic calls. The `.RTN` command also automatically resets the interval to the one in effect when the ASF sequence started. You can, however, add an NR argument to the `.RTN` command (.RTNNR) to direct OMEGAMON for IMS to keep the new interval in effect.

Because the **SL=** parameter is used, ASF automatically turns on the REPORT log when the exception occurs (if it is not already on). When you leave automatic mode, ASF also turns off the log (if it is in the off state before the exception triggers).

When you use ASF to turn on the log automatically with **SL=**, OMEGAMON for IMS first logs the screen in use, and then branches to the scheduled screen space. Any exceptions can be logged before the ASF sequence begins.

Chapter 15. The Database Control (DBCTL) environment

Under a DBCTL environment, you can access both DL/I full function databases and fast path data entry databases (DEDBs) through a Coordinator Control (CCTL) system.

OMEGAMON for IMS differences in DBCTL

When you are using OMEGAMON for IMS in a DBCTL environment, different supported features and functions exist.

- A DBCTL system has no Data Communications (DC) support. Therefore, transactions or message queues do not exist, and no support exists for VTAM, MSC, ISC, and MFS.
- In the absence of DC support, AF/REMOTE is not available if you are running in a DBCTL environment.
- Some DEXAN displays might look different from ones that you are familiar with when you work in an environment other than DBCTL. (Since transactions do not exist in a DBCTL environment, fields about transactions do not display.)

OMEGAMON for IMS command differences in DBCTL

In a DBCTL environment, the following OMEGAMON for IMS major and immediate commands are not applicable.

ASYS
AUTO
BLGx
CIOP
CLSx
CNVx
DEDP
DIRx
FMTx
FREP
FSTP
HIOP
IRTA
ITX
LNEx
MFP
MPP
MSDx
NODx
PTRx
QBUF
RCDx
RECA

RGNF
RGNM
SPAP
TCOx
TRMx
TRPQ
TRXx
TXC
TXPQ

Exception differences in DBCTL

The following exceptions are not applicable in a DBCTL environment.

ACIO
ACIS
ACIW
ACIX
ACWA
ACWS
ACWW
ACWX
AFRE
AHIO
ALMD
AMFS
AMSG
AQBD
ARAU
ARCB
ARSP
ASMD
BLGH
CMHI
CMLO
CVAH
CVHI
CVSH
FCIO
FCWA
FHIO
FMFS
IMHI
IMLO
ITWH
LMGH
LTOQ
MDHI
MDLO
MFSH
MIRT
MPCH

MSDI
MSDO
NACB
NDRE
NILU
NLOQ
NSDC
NTIQ
NVAP
PIMC
PIRP
PROQ
QBKH
ROHI
ROLO
SAPW
SCTR
SDSP
SMGH
SPAH
TCOI
TCOT
TMFH
TXIQ
WMHI
WMLO

Minor command differences in DBCTL

The following major commands contain minor commands that are not applicable in a DBCTL environment.

DUMP

$DIR
BALG
BHDR
CCB
CLB
CNT
CTB
ECNT
FIDX
MFP
QBUF
RCTE
SMB

PSBx

ARVL
PRCS
TX#

RGNx, BMP, and RDB2

MGN

MGU
MIRT
MNUM
MPRG
MSDC
MSGT
OCUP
PLIM
PLSC
QTME
SCLC
TERM
TIME
TPCB
TRAN

IDEG

DOPT

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 Knowledge Center 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 Knowledge Center 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.

Product legal notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

IBM Corporation 2Z4A/101 11400 Burnet Road Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations may not appear.

Trademarks

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

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

Microsoft, Windows, and Windows NT 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.

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.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name for purposes of session management, authentication, and single sign-on configuration. These cookies cannot be disabled.

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

Numerics

A

B

C

D

E

I

J

K

L

P

Q

R

S

T

U

V

Product Number: 5698-T02