

IBM OMEGAMON for z/OS
5.5

OMEGAMON for MVS User's Guide



Note

Before using this information and the product it supports, read the information in [“Notices” on page 319.](#)

Edition notice

This edition applies to version 5, release 5, modification 0 of IBM OMEGAMON for z/OS (product number 5698-T01) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1992, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures.....	ix
Tables.....	xiii
Chapter 1. OMEGAMON for MVS overview.....	1
The logical tuning approach.....	1
OMEGAMON menus.....	2
User Profile Facility.....	4
Quick access method.....	5
Help screens.....	5
OMEGAMON menu PF key definitions.....	7
Exit.....	8
OMEGAMON command mode.....	8
Command types.....	8
Command format.....	9
Help for major, minor, and immediate commands.....	10
Help for INFO-line commands.....	11
INFO-line format.....	12
Screen display symbols.....	13
Comment lines (>).....	13
Continuation character (+).....	14
Input character (:)......	14
Screen control.....	14
Up and down scrolling.....	15
Clearing the screen.....	15
Deleting comment lines.....	15
Inserting and deleting lines.....	16
Displaying continuation lines of command output.....	16
Delaying or holding automatic updating.....	17
Chapter 2. Configuring and running in different modes.....	19
Dedicated mode.....	19
Customizing for dedicated mode.....	19
Starting OMEGAMON in dedicated mode.....	20
Stopping OMEGAMON in dedicated mode.....	21
Using secondary dedicated terminals.....	21
Running multiple dedicated sessions.....	21
Issuing OMEGAMON commands from the system console.....	22
Security in dedicated mode: the re-logon feature.....	22
VTAM mode.....	23
Customizing the VTAM startup procedure.....	23
VTAM mode optional features.....	25
Starting OMEGAMON in VTAM mode.....	25
Stopping OMEGAMON in VTAM mode.....	27
TSO mode.....	27
Customizing for TSO mode.....	27
Starting OMEGAMON in TSO mode.....	28
Stopping OMEGAMON in TSO mode.....	29
Modifying virtual terminal pool definitions.....	29

ISPF mode.....	29
OMEGAMON ISPF mode components.....	29
Configuring IBM OMEGAMON for z/OS to run in ISPF mode.....	30
Starting IBM OMEGAMON for z/OS in ISPF mode.....	31
Stopping OMEGAMON in ISPF mode.....	32
Modifying virtual terminal pool definitions.....	32
Authorizing OMEGAMON libraries for TSO and ISPF modes.....	33
Chapter 3. Operational commands.....	35
Command modifiers and help symbols.....	35
Rate, difference, and percentage arguments.....	35
Operational INFO-line and immediate commands.....	36
Chapter 4. Screen spaces.....	71
Invoking screen spaces.....	71
Creating or modifying screen spaces.....	71
Loading screen spaces.....	73
Listing screen spaces.....	73
Renaming a screen space.....	73
Deleting a screen space.....	74
Invoking screen spaces automatically.....	74
Conditional screen space fetching.....	74
Delayed screen space fetching.....	74
Looping screen spaces.....	75
Testing .FGO routines.....	75
Manipulating the log within screen spaces.....	75
Using screen spaces with the /ZOOM command.....	75
Chapter 5. The User Profile Facility.....	77
Multiline input.....	79
Profile maintenance commands.....	79
Controlling session and display options.....	80
Setting print output options.....	80
Setting color options.....	85
Setting operational parameters.....	88
Exclude Enqueue Names command.....	90
Chapter 6. Exception analysis.....	93
Controlling exception analysis.....	93
Starting exception analysis.....	94
Defining exception characteristics.....	95
Summarizing exception activity.....	100
Defining exception characteristics.....	101
Controlling groups.....	102
Defining groups or mask entries.....	103
Setting special thresholds.....	111
Exceptions.....	111
Swap reasons.....	117
Enqueue waits.....	117
Device waits.....	118
Miscellaneous waits.....	118
Unresolved waits.....	118
Chapter 7. General system information.....	125
System information.....	125
System information major command.....	125
System information minor commands.....	126

CPU utilization commands.....	145
System information commands.....	153
Paging immediate commands--multiplot.....	159
Page data set information.....	161
Page data set information major command.....	161
Page data set information minor commands.....	161
Enqueue information immediate commands.....	162
Data set information immediate commands.....	165
Control block displays.....	167
Control block display major command.....	167
Control block display minor commands.....	167
LPA modules immediate command.....	171
GRS immediate command.....	172
Workload Manager.....	172
Chapter 8. CSA Analyzer.....	177
Configuring the CSA Analyzer.....	177
Changing the startup parameters.....	177
Startup parameter symbolics.....	178
Implementing CSA Analyzer IPCS support.....	178
CSA Analyzer commands.....	179
CSAA immediate command.....	179
CSAF immediate command.....	181
CSA Analyzer-related exceptions.....	182
CSA Analyzer-related minor commands.....	183
Chapter 9. Device information.....	185
Device listing commands.....	185
.DMP command.....	186
Disk information.....	186
Disk select major commands.....	187
Disk space immediate command.....	191
Selected disk minor commands.....	191
Examples of disk commands.....	195
Disk I/O information.....	196
Tape information.....	197
Tape select major commands.....	197
Long-term device utilization.....	198
Long-term device utilization major command.....	198
Long-term device utilization minor commands.....	199
Long-term channel path utilization.....	205
Long-term logical control unit (LCU) utilization.....	205
Chapter 10. Address space information.....	207
Address space information major commands.....	207
Address space information minor commands.....	210
Address space identification.....	210
Address space environment.....	211
Address space storage activity.....	214
Address space CPU activity.....	216
Address space SRM activity.....	218
Address space time information.....	221
Miscellaneous.....	222
Chapter 11. Authorized commands.....	225
Using authorized commands.....	225
Supplying the password.....	225

How to use the action character.....	225
OMEGAMON SRB/XMS facility.....	226
Address specification for storage commands.....	228
Displaying and modifying storage.....	229
Storage display commands.....	229
Storage scan commands.....	231
Storage modification commands.....	236
Displaying and modifying data space and hiperspace storage.....	237
Console-related commands.....	242
Collecting data about address spaces.....	245
Terminating an address space (KILL).....	256
Find users of a device (FNDU).....	257
SEEK contention analysis.....	259
Address space minor commands.....	261
Program library facilities.....	263
Chapter 12. Degradation Exception Analyzer (DEXAN).....	267
How DEXAN works.....	267
How to use DEXAN commands.....	268
DEXAN display commands.....	268
Controlling the DEXAN data collector.....	270
Job degradation analysis.....	273
Overview of job analysis data collection.....	273
Overview of job analysis data display.....	273
Job analysis control commands.....	273
Job degradation display commands.....	276
System-wide analysis.....	278
Displaying system-wide degradation data.....	278
Controlling the DEXAN data display.....	280
Execution state summary.....	282
Chapter 13. Impact analysis.....	287
Workload contention.....	287
Selecting a diagnostic tool.....	287
Impact analysis displays.....	288
Solving critical problems.....	288
Types of contention.....	288
Performing impact analysis functions.....	290
IANL command syntax.....	290
Starting and stopping an impact analysis session.....	292
Selecting a workload.....	292
Defining a group workload.....	292
Selecting a user-defined group.....	293
Listing workloads and groups.....	293
The impact analysis display format.....	294
How resource contention is shown.....	297
The Workload Impact Profile.....	298
The impact analysis time-out facility.....	299
Setting defaults.....	299
Chapter 14. Applications tuning.....	303
Applications tuning facility INSP major command.....	303
Applications tuning minor commands.....	304
ICPU minor command.....	304
MLIB minor command.....	307
Appendix A. Automating and logging features.....	309

Controlling automating features.....	309
XLF commands.....	309
ASF commands.....	310
TSF commands.....	310
Setting parameters for XLF and ASF operation.....	310
Setting parameters for TSF operation.....	311
Logging parameters for XLF, ASF, and TSF.....	313
Using screen spaces in ASF and TSF.....	313
How ASF and TSF screen spaces execute.....	313
Using the .RTN command.....	313
ASF example.....	314
Support information.....	317
Notices.....	319
Trademarks.....	320
Privacy policy considerations	320

Figures

- 1. Initial OMEGAMON menu screen..... 2
- 2. Exceptions menu..... 3
- 3. Typical exceptions display..... 4
- 4. Profile Maintenance and Session Controls menu..... 5
- 5. Profile Facility Help screen.....6
- 6. General Help screen..... 7
- 7. Exit Confirmation screen..... 8
- 8. INFO-line command help.....12
- 9. Stopping an OMEGAMON for MVS session..... 29
- 10. Invocation Menu displayed by KOMSPF CLIST..... 32
- 11. Sample CSECT to add OMEGAMON to TSO authorized list.....33
- 12. Typical SMF display.....59
- 13. Result of /STK2.....61
- 14. A sample screen space..... 72
- 15. Screen space list..... 73
- 16. Typical enqueue name table..... 90
- 17. Typical output from the XQN ADD command..... 91
- 18. Typical XQN DEL command output..... 91
- 19. Partial LEXC display..... 95
- 20. Typical XACB display..... 99
- 21. Partial XACB terse mode display..... 99
- 22. Partial XSUM display..... 100
- 23. Summary display of current address space threshold table..... 104

24. Output from the ASG DIS command.....	105
25. Typical output from ASG ADD command.....	106
26. Typical output from ASG DEL command.....	107
27. Typical missing task table.....	109
28. Typical output from MTA ADD command.....	110
29. Typical output from MTA DEL command.....	111
30. Use of plot symbol.....	126
31. Output of CSA command.....	127
32. Typical CSAR output.....	128
33. Typical ECSA output.....	128
34. Typical ESQA output.....	129
35. Typical output of RMAP command.....	131
36. Typical output of RMLF command.....	132
37. Typical output for SQA command.....	132
38. Typical output from VMAP command.....	133
39. Typical output for SRM command.....	138
40. ENV command output with HiperDispatch enabled.....	141
41. Typical output for RMSU command when LPAR is not in an LPAR group.....	143
42. Typical output for XRMSU command if LPAR is not in an LPAR group.....	143
43. Typical output for XRMSU command if LPAR group capping is in effect.....	144
44. Typical output display for MCPU with APF-authorization.....	146
45. Typical output for SCPU command.....	152
46. Typical output for HDSP command.....	155
47. Typical output for LPAR command.....	157
48. Typical output for XLPAR command.....	158

49. Typical output for YLPAR command.....	159
50. Output for WLM with SDPD and SRVC minor commands.....	174
51. Output for SRVC with period and service class name specified.....	175
52. CSAA USAGE sample output.....	180
53. CSAA TREND sample output.....	181
54. CSAA OWNER sample output.....	181
55. CSAF sample output.....	182
56. Output of SVOL command for EAV.....	191
57. Format of long-term device utilization minor commands.....	199
58. Typical minor command of STAT display.....	199
59. Typical minor command of STATP display.....	200
60. Typical LCU display.....	206
61. OCMD and CONS Output.....	244
62. AMAP display format.....	247
63. Typical LSQA results screen.....	251
64. SEEK major command output.....	259
65. DATA minor command output for all observed SEEKS.....	260
66. PLOT minor command output.....	261
67. Typical resource impact profile for address space.....	269
68. Collection bucket descriptions (/BLST).....	281
69. Impact analysis display format.....	294

Tables

1. Dedicated mode parameters.....	20
2. VTAM mode parameters.....	24
3. TSO mode parameters.....	28
4. Profile definition.....	77
5. OUTF minor commands and initial values.....	81
6. Color and highlighting settings in .SCC.....	88
7. Exception Analysis control commands.....	93
8. CSA Analyzer start up options and default values.....	177
9. CSAA function keywords.....	179
10. CSAA filter keywords for limiting displays.....	180
11. Commands that require the action character.....	225
12. Commands used with IPRO.....	269
13. Data analysis parameters.....	280
14. List of SRM swap reasons.....	283
15. List of miscellaneous wait reasons.....	283
16. List of device wait reasons.....	285
17. List of enqueues.....	286
18. Commands to set impact analysis defaults.....	300

Chapter 1. OMEGAMON for MVS overview

OMEGAMON® for MVS™ is an autonomous component of OMEGAMON for z/OS. It provides real-time performance data on the z/OS® operating system to IBM OMEGAMON for z/OS that can be viewed using the Tivoli® Enterprise Portal interface. It can also be accessed directly through a 3270 interface. OMEGAMON for MVS does not require hooks, Supervisor Call instructions (SVCs), or other changes to the operating system.

OMEGAMON is used primarily in two ways:

1. Operations staff uses the OMEGAMON automatic exception analysis to check potential trouble spots within z/OS every few seconds. Exception analysis displays messages that warn of existing or impending hardware and software problems from both a system and an individual address space perspective. OMEGAMON can trigger exception messages when exception values exceed the thresholds you set, or when system conditions do not comply with your installation's standards. This section describes all of these analyses in detail.
2. Systems programmers can run OMEGAMON as a TSO transaction or as a VTAM® application to use commands that display the internals of z/OS in real time. Systems programmers can customize OMEGAMON by selecting the commands that display only the information they need to see.

OMEGAMON displays information on such topics as:

- CPU utilization by address space
- System resources manager (SRM) parameters
- Page I/O service times
- Logical channel queues or control units
- I/O contention by device, control unit, or channel
- Workload Manager (WLM)

This section describes the features and commands used in OMEGAMON for MVS. Reference information for OMEGAMON major and minor commands is included, along with descriptions of the following features:

User Profile Facility

Controls the characteristics of an active OMEGAMON session.

Exception analysis

Produces a warning message when a parameter goes above or below a specified threshold.

CSA Analyzer

Feature that analyzes and displays overall common storage usage, displays high level graphic displays on usage trends, and reports on growth at the job or system level.

Bottleneck analysis

Shows a breakdown of response time into active states, idle states, and delayed states. Bottleneck analysis includes the following:

Degradation Exception Analyzer (DEXAN)

Feature that analyzes degradation (unproductive time or the time a workload spends waiting for a system resource, rather than actively using the system).

Impact analysis

An optional feature that identifies the source of workload contention.

The logical tuning approach

All of the features and facilities of OMEGAMON are designed around the concept of a logical tuning approach for improving the performance of your z/OS system.

The logical tuning approach methodology consists of the following tasks:

- Setting standards for z/OS performance at your site

You set standards for z/OS performance via OMEGAMON exception analysis. [Chapter 6, “Exception analysis,”](#) on page 93 explains procedures for setting exception analysis thresholds.

- Monitoring z/OS to measure actual performance problems against these standards

Monitoring of actual z/OS performance consists of asking these two basic questions:

- Is response time adequate?
- What are the problems in the system?

- Identifying the cause of performance problems

When response time falls below your site’s performance standards, you can use bottleneck analysis to help you identify the cause of the problem. With the information you receive from bottleneck analysis, you can take action to solve the problem.

- Taking action to correct problems

Some of the system problems that exception analysis messages indicate can also be identified and solved with bottleneck analysis; others call for the resource analysis available with the various OMEGAMON exploratory commands.

OMEGAMON helps you carry out each of these performance tuning steps.

OMEGAMON menus

OMEGAMON contains a menu facility that provides access to all functions of OMEGAMON and related products.

The Main Menu screen is the OMEGAMON screen that follows the copyright screen. To access the Main Menu screen, type MENU on the INFO-line and press Enter.

After you press Enter, you will see a menu similar to the following figure (unless your site has modified the initial menu screen):

```

----- ZMENU   VTM      OM/DEX   V420./I SYS 02/13/09 14:00:20
> Help/News PF1  Exit PF3   Keys PF5   Colors PF18  Cmd mode PF24
>
>          Enter a selection letter on the top line.
=====
>          OMEGAMON MAIN MENU
=====
_ E EXCEPTIONS ..... Current system problems
_ J JOB ANALYSIS ..... TSO, BATCH, STC information
_ R RESPONSE TIME .... TSO network and host response times
_ U USER ..... Analyze a single user or task
_ B BOTTLENECKS ..... Pinpoint bottlenecks in TSO, BATCH, STC

_ I I/O ..... DASD, tape, channel, and LCU information
_ S STORAGE ..... Real and virtual storage use
_ O COMMON STORAGE ... Monitor and analyze common storage usage
_ C CPU ..... LPAR, processor and workload utilization
_ M MVS ..... System information, SRM control information
_ A ACTION ..... Operator tools, action commands

_ P PROFILE ..... Profile Facility and session options
>          *** USE PF18 TO CONTROL DISPLAY COLORS ***
=====

```

Figure 1. Initial OMEGAMON menu screen

IBM® sometimes enhances these menus between releases, so your own screen may differ somewhat from this example.

If your site has modified the initial screen, you might not automatically come to this menu when you start OMEGAMON. You can, however, reach it at any point by typing MENU on the first line (INFO-line) over the underscores and pressing Enter.


```
MENU_____ ZMENU   VTM   OM/DEX   V420./I SP22 02/13/09 14:00:20
```

This menu contains several performance monitoring and tuning options. The available PF key definitions are listed before the options.

To select a menu option, type the option letter on the INFO-line starting over the first underscore (_), and press Enter.

For example, to access the EXCEPTIONS option, type E on the first line of the screen and press Enter. Select EXCEPTIONS now, and the following menu displays:

```
_____ ZEXCP   VTM   OM/DEX   V420./I SP22 02/13/09 14:04:11
> Help PF1                               Exit PF3
>                                     Type your selection on the top line and press ENTER.
=====
>                                     EXCEPTION INFORMATION
- A OPS ..... Operation status: CPU, active BATCH, TS0, exceptions
- B SYSTEM ..... System wide exceptions
- C HISTORY ..... History of all tripped exceptions

> To control exception analysis processing and modify thresholds,
> select option P from the Main Menu.
```

Figure 2. Exceptions menu

The Exceptions Menu gives you access to several options related to z/OS system exceptions. For example, the first option, OPS, displays all operation status exceptions.

Select the OPS option by typing A on the INFO-line and pressing Enter. When there are no exception conditions in your system, OMEGAMON displays the following message:

```
+      '==> *** NO EXCEPTIONS TRIPPED ON THIS CYCLE *** <==      '
```

When exceptions exist, you will see a screen similar to the one in [Figure 3 on page 4](#).

```

----- ZOPS      VTM      OM/DEX  V420./I SP22 02/13/09 14:05:36
> Help PF1      Back PF3      Up PF7      Down PF8      Zoom PF11
-----
>
                        OPERATION STATUS
-----
SCPU10  CPU Utilization  0__10__20__30__40__50__60__70__80__90__100
+      Total          10.00 |---1> . . . . . . . . . . . . . . . . . . . .
-----
BATX  IMS9DMS1 IMS9DFP2 IMS9DFP3 IMS9DFP1 IMS9ZMS1 I91DCCON IMS9ZFP1 IMS9ZFP3+
step  IMS9DMS1 IMS9DFP2 IMS9DFP3 IMS9DFP1 IMS9ZMPP I91DCCON IMS9ZFP1 IMS9ZFP3
elap  3:08 DY  3:08 DY  3:08 DY  3:08 DY  1:09 DY  2:03 DY  1:09 DY  1:09 DY
-----
>      TSO users
#TSOJ          7
-----
> For more information, place the cursor on the exception name and press PF11.
LXGRPHD  OMEGAMON/MVS Group Exception Analysis
LXGRPOP  OMEGAMON/MVS Group Exception Analysis
+ XREP Number of Outstanding Replies = 5
+ FXFR STC *MASTER*      | Fixed Frames in use = 3278
+ FXFR STC PCAUTH        | Fixed Frames in use = 196
+ WAIT  PCAUTH           | Wait: 5:00 DY
+ FXFR STC TRACE         | Fixed Frames in use = 866
+ WAIT  TRACE            | Wait: 2:19 DY
+ FXFR STC M5GBHUB1      | Fixed Frames in use = 338
+ WAIT STC K3HDI22       | Swap: 6:02 HR Long-Wat
+ WAIT STC L3IAM2W9      | Wait: 33 SEC
+ WAIT STC L3IAMCW9      | Wait: 37 SEC
+ WAIT STC L3IAETW9      | Wait: 5:08 MN
+ WAIT BAT I91DCCON      | Wait: 43 SEC
-----
SXQCB   M5S042H1 Wait Exc KLVGLOCK OM542DEM
+       M5GBHUB1 Wait Exc KLVGLOCK PKDSPXGR
-----
TPMT
dusr
dvol
-----

```

Figure 3. Typical exceptions display

User Profile Facility

The Profile Facility allows you to customize certain characteristics of your OMEGAMON session. You can save the option settings in a profile for subsequent sessions by selecting the SAVE/DELETE option or by pressing PF22. For example, you can set exception analysis thresholds, or the display colors for your session with the Profile Facility. If you save the settings you have specified, you can use them again in future OMEGAMON sessions.

To go to the Profile Facility Menu, type P from the Main Menu and press Enter. OMEGAMON displays the menu shown in [Figure 4 on page 5](#).

```

----- ZOPT      VTM      OM/DEX  V420./I SP22 02/13/09 14:20:33
> Help PF1          Back PF3          Save Profile PF22
>      Type a selection letter on the top line and press Enter.
=====
>      PROFILE MAINTENANCE AND SESSION CONTROLS
=====
>      Profile Maintenance
_ A EXCEPTIONS ..... Set exception analysis thresholds
_ B CONFIGURE ..... Set default startup configuration
_ C MISSING TASKS .... Missing task analysis
_ D GROUPS ..... Set address space threshold groups
_ E EXC CONTROLS ..... Other controls for exception analysis
_ F NAMES ..... Domain, performance group, enqueue exclude names
_ G SAVE/DELETE ..... Create, delete, save profiles

>      Quick-Set Controls
_ H COLOR ..... 7 color, 4 color, monochrome, set colors
_ I BACKGROUND ..... Set background (XLF) processing (DEXAN users only)
_ J AUTO ON ..... Turn automatic updating on (VTAM mode only)
_ K AUTO OFF..... Turn automatic updating off (VTAM mode only)
_ L LOGGING ..... Log on, log off, spin out log

> Settings you change through this menu become permanent when you save
> your profile. To do this, press PF22 from any screen.
=====

```

Figure 4. Profile Maintenance and Session Controls menu

Quick access method

OMEGAMON provides you with a quick way to select menu options that you use frequently.

To go directly to a menu screen from any other screen, type on the INFO-line the letter corresponding to the menu selection followed by a period. For example, to reach the Exceptions Menu, type E. on the INFO-line and press Enter. To go directly to the Operation Status screen, type E.A on the INFO-line of any screen and press Enter.

For example,

Important: Do not use screen space names to navigate in the menu system. The menu system is designed to permit navigation by PF keys and single letter commands only. If you do use screen space names for navigation in the menu system and then want to use PF keys, you must press PF4 to return to the Main Menu, which will reinitialize the PF key settings.

Help screens

OMEGAMON provides help screens that give you a context-sensitive explanation of any OMEGAMON menu or screen display when you press PF1.

For example, if you are in the Profile Maintenance and Session Controls Menu and you want help, press PF1 to see a display similar to the following figure:

```

-----ZOPTH   VTM   OM/DEX   V420./I SP22 02/13/09 14:22:14
> General Help PF1   Back PF3   PF Keys PF5   Up PF7   Down PF8
-----
>
> PROFILE MAINTENANCE AND SESSION CONTROLS -- HELP
>
> This menu allows you to dynamically control your session options. For users
> of past releases, it replaces the OMUSER module.
>
> OMEGAMON ships a default profile, so you may start up your session
> immediately. If you are the product installer, you may customize a profile
> and save it as the installation profile; this will then be the default
> profile for everyone at your site.
>
> As an OMEGAMON user, you may also create and save your own individual user
> profile. You do so by:
>
> * Selecting the paths on this menu that contain the settings
> you want to change.
> * Setting the values as appropriate for your needs.
> * Then, saving a new profile by selecting PROFILE MAINT on this menu.
>
> To change an option for your current session, select the option from the
> menu, find the setting you wish to change, and type the new setting over the
> old setting. The change takes effect immediately for the duration of the
> current OMEGAMON session (for all but startup options). To make the change
> permanent, save your profile by selecting the SAVE/DELETE option on this
> menu or pressing PF22 from any screen.
>
> These menus also provide quick access for some options that are frequently
> changed, such as color or logging. In these cases, the menu selection
> activates the option for you. Afterwards, you will see a confirmation
> message at the bottom of the menu. These changes are in effect for the
> duration of this session only, unless you save your profile with the
> SAVE/DELETE option or PF22.
>
> For an explanation of what each option means, select the option and then
> press PF1 for help.
>
> Please note that for the Profile Facility to function, you need a file
> allocated to DDNAME RKOMPROF. If you wish to save profiles, you need a file
> allocated to DDNAME RKOMPFSV. Both RKOMPROF and RKOMPFSV must be partitioned
> datasets and must be blocked VB with LRECL=3332 and BLKSIZE=3336.
=====

```

Figure 5. Profile Facility Help screen

To view additional help, press PF1 again from any help screen for information about the various OMEGAMON general help facilities.

For example, press PF1 from the previous help screen to display a general OMEGAMON help menu similar to the following one:

```

----- ZHELPM  VTM  OM/DEX  V420./I SP22 02/13/09 14:24:16
> Back PF3                                     Main Menu PF4
> Type your selection on the top line and press Enter.
=====
> OMEGAMON GENERAL HELP MENU

_ A NEWS ..... What's new in this release of OMEGAMON
_ B MENUS ..... OMEGAMON's menu system
_ C EXCEPTIONS ..... Recommendations for resolving exception conditions
_ D STATUS MONITOR ... How to access Status Monitor display
_ E ZOOMING ..... More detail or related information
_ F PF KEYS ..... PF keys in the menu system
_ G CUSTOMIZE ..... Customize a menu path for your site
_ H COMMANDS ..... Command mode and command help
_ I SUPPORT ..... IBM customer support

=====

```

Figure 6. General Help screen

From a general help screen, press PF3 to return to the previous help screen. Press PF3 again to return to the screen from which you initially requested help.

Exception recommendations

OMEGAMON offers a special kind of help screen for exceptions, called a recommendation screen.

These screens offer suggestions for solving the problem that an exception indicates. To view a recommendation screen, move the cursor to the name of the exception and press PF11.

OMEGAMON menu PF key definitions

The PF key definitions supplied for use within the OMEGAMON menus are listed in the following table.

Following is a list of PF keys and their descriptions.

PF Key	Description
PF1/PF13	Help
PF2/PF14	Unused
PF3/PF15	Back to previous screen
PF4/PF16	Main Menu
PF5/PF17	List PF keys
PF6	Print the screen
PF18	Set colors
PF7/PF19	Scroll up
PF8/PF20	Scroll down
PF9/PF21	Unused
PF10/PF22	Save profile
PF11/PF23	Zoom for detail
PF12/PF24	Menu/Command mode switch

Exit

To stop OMEGAMON at any time, type an X on the INFO-line and press Enter.

The following screen displays, which allows you to return to the menu system, exit the menu system to command mode, or end the OMEGAMON session:

```
----- ZQUEST  VTM    OM/DEX  V420./I SP22 02/13/09 14:26:26
> Help PF1                      Exit PF3
>                               Type your selection on the top line and press ENTER.
=====
>                               EXIT  CONFIRMATION
=====
_ M MENU ..... Return to menu system
_ C COMMAND ..... Exit menu system to command mode OMEGAMON

_ X EXIT ..... End OMEGAMON session

> If you leave the menu system and enter command mode, use PF12 or PF24
> to return to the menu system.

> If you enter X from anywhere in OMEGAMON (except this screen), you
> return to this menu.
=====
```

Figure 7. Exit Confirmation screen

To confirm your choice and stop OMEGAMON, type X on the INFO-line and press Enter.

OMEGAMON command mode

To operate OMEGAMON in command mode, you need to know the types of OMEGAMON commands and the appropriate time and place to enter each type.

Command types

OMEGAMON supports four types of commands.

The following figure shows the four types of commands used in OMEGAMON along with some sample output.

```
/PRINT_____#01  TSO    OM/DEX  V420./C SYS  02/12/09 17:03:37  5B
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 +
```

The following list describes the four types of commands:

INFO-line

These commands perform control functions such as printing a screen (/PRINT) and stopping your OMEGAMON session (/STOP). INFO-line commands are executed first and, unlike the other types of commands, disappear as soon as they execute. Therefore, you cannot save them in a screen space. INFO-line commands always begin with a slash (/), and must be entered on the first line, starting 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 utilization, or storage utilization. In the example, the major command DISK produces a list of online disks. You can enter major commands on any line following 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 listed with DISK. You can enter minor commands on any line following the INFO-line, but they will not execute 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. In the example, .MIN produces a list of all the minors of DISK. Immediate commands can also provide screen and session controls, and often perform the same or similar functions as INFO-line commands of the same name. This enables you to include control commands as part of a screen space. Enter immediate commands on any line following the INFO-line. (It is permissible to enter immediate commands between a major command and one of its minors.)

Command format

Here is another example of OMEGAMON commands and their output. Notice that command fields are four characters, but commands can be entered with labels or arguments. The diagram that follows the screen display shows the command entry area.

```
.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  +
```

```
lccorn _____
      ────> extended argument or keyword field (cols. 8 through 72)
      ────> argument field (cols. 6 and 7)
      ────> command name (cols. 2 through 5)
      ────> label field (col. 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 (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.

command name

This field contains the command name and is four characters in length. Although the command name field consists of columns 2 through 5, OMEGAMON will recognize most commands begun in column 1 and will automatically shift the command one space to the right.

argument field

Many commands accept arguments that modify their function or specify output options. In the example, the .MJ command produces a list of major commands. It has been entered with an argument of DS, a group code which specifies that only disk major commands be listed.

extended argument

Arguments are entered in columns 8 through 72. In the example, the GDEV command has been entered with 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. Note that a parameter string cannot be extended onto a second line. The command (along with keywords) must be re-entered on succeeding lines.

Help for major, minor, and 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 (+) to the right of the one-line help indicates that more (extended) help is available.

?

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

/

Displays an extended help that stays on the screen until you clear it. You can clear the help text with the .DCL command.

;

Displays an extended help that disappears from the screen on the next cycle.

Command help examples

Examples of these three symbols used with the .WAT command follow.

This entry for the .WAT immediate command

```
? .WAT
```

produces a one-line help as shown in the following display:

```
----- ZMENU   VTM   OM/DEX   V420./I SYS 02/13/09 14:43:53 23
>.WAT
>   Specifies a delay before executing the commands that follow.      +
```

When you use a slash instead of a question mark,

```
/.WAT
```

an extended help is displayed:

```
>.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.
```


The entry

```
; .WAT
```

produces the same output as /.WAT, except that plus signs display in column 1 beneath the one-line help. The plus signs are continuation symbols and indicate that the extended help text will disappear on the next cycle.

You can obtain help for a minor command by using the symbols 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 displays a help screen similar to the following one:

```
?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).                +
```

In cases where the same minor command name is used for multiple major commands, but the function and help text differs, OMEGAMON lists the possibilities and instructs you to place your cursor under the desired major command name. When you press Enter, the desired help appears.

If you know that the minor command is shared among several major commands, you can type the desired 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.

Help for INFO-line commands

INFO-line command help is obtained using the .ILC immediate command rather than with a symbol in column 1.

Enter the name of the INFO-line command (cccccc) in this format (the slash is not required):

```
.ILC ccccc
```

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

```
.ILC DEF
```

You will see a help like that in [Figure 8 on page 12](#).

```

>.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

INFO-line format

The first line of an OMEGAMON screen display is called the INFO-line. The INFO-line accepts input (such as the /PRINT command) and displays status information about your session.

The INFO-line looks similar to the following example:

```
/PRINT_____ #01 TSO OM/DEX V420./C SYSA 02/12/09 17:03:37 5B
```

The last eight positions of the input line may be reserved by many commands and may not be used to input data.

The components of the INFO-line using the variables are shown in the following figure, using variables:

```
<input>_____ccccccc mode LOG pr V420.cc sysid mm:dd:yy hh:mm:ssnnnc
```

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 pre-defined screen is invoked when the screen space name is entered. OMEGAMON distinguishes screen space names from INFO-line commands because INFO-line commands always start with a slash (/). They are explained in the next section.

ccccccc

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

mode

Shows the type of session or mode being used to access OMEGAMON. The possible values are:

DED

A dedicated mode session

TSO

A TSO mode session

SPF

An ISPF mode session

VTM

A VTAM mode session

LOG

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

pr

Displays a code or multiple codes for the product(s) running. For example, this variable will be OM for OMEGAMON for MVS.

Note: If DEXAN is installed, an asterisk (*) appears after the DEX product code when the DEXAN collector is running.

V420

Indicates the product level. The figure shows Version 420.

cc

Identifies the user profile in use.

sysid

Reflects the system identifier that you assign with the SYS= startup parameter. This value is used to identify cross memory and cross system directors and collectors. The default is the SMF ID.

mm:dd:yy

Indicates the date the screen was last refreshed.

hh:mm:ss

Indicates the time the screen was last refreshed.

nnnn

Indicates the scrolling depth.

c

Indicates whether the Automatic Screen Facility (ASF) is currently active. The variable c can be one of these values:

A

Indicates that the ASF is currently active.

B

The terminal bell has been activated and will sound if exceptions occur.

S

Screen spaces are currently stacked.

Note: S appears only if A is not true. B appears only if neither A or S are true

Screen display symbols

OMEGAMON displays several screen symbols with command output.

- [“Comment lines \(>\)” on page 13](#)
- [“Continuation character \(+\)” on page 14](#)
- [“Input character \(:\)” on page 14](#)

Comment lines (>)

Any command entered on that line will not execute.

To designate a comment line, enter a greater-than sign (>) in column 1.

In addition to the comment symbols that you enter yourself, OMEGAMON automatically comments out some commands after they execute to prevent re-execution on the next cycle. OMEGAMON also comments out help text, so that it remains on the screen until you clear it.

Continuation character (+)

So that the display does not shift off the bottom of the screen, OMEGAMON displays only one line of data at a time unless you request additional lines. If more output is available, OMEGAMON displays a plus sign (+) at the end of a line of output. Online help text also displays the continuation character when more help is available.

You might see this continuation character after you issue certain major commands that select a long list of items (for instance, all online disks). [“Displaying continuation lines of command output” on page 16](#) explains how to request the additional lines.

Commands that produce multiple display lines when they execute display the plus sign (+) in column 1. For example, the .VTM command displays multiple lines of information on VTAM users.

```
.VTM Userid Terminal Mode Session Start Last Update
+ TS0X07 L65F VTM 02/11/09 11:45:32 02/12/09 13:14:55
+ TS0X06K BBLVM06 VTT 02/12/09 13:57:32 IN INITIALIZATION
+ AFOPER BBLVM04 VTT 02/12/09 11:43:22 02/12/09 13:57:12
+ TS0X21 L616A09 VTS 02/12/09 10:23:31 02/12/09 11:24:31
+ TS0X36A L674 VTM 02/12/09 13:40:33 02/12/09 13:57:51
+ TS0X04 L655 VTM 02/12/09 11:35:32 02/12/09 12:14:55
```

If you have more commands entered following a multiple display line command, they may shift off the bottom of the screen. These commands reappear as soon as you scroll down through the output.

If you enter a new command on a continuation line, OMEGAMON automatically moves the new command beneath the existing output lines.

Input character (:)

Some OMEGAMON commands allow multiline input for setting parameters. If you issue such a command, OMEGAMON responds with a table of keywords and their current settings. On each line of a multiline input command, column 1 displays a colon (:) indicating that you can overwrite the current setting to change the value.

For example:

```
----- ZMENU TSO LOG OM/DEX V420./C SYS 02/12/09 10:32:22 5
.SET
: FGOLIMIT = 64 FGLOOP = OFF
: GDEVUCBS = 200 INTERVAL = 10.00
: IODELAY = 5 LOOPCOUNT = 15000
: LOOPTIME = 5.00 PAGELIMIT = 400
: PEEKSIZE = 4096 STATUSMODE = OFF
```

Screen control

This section describes the following screen control functions:

- [“Up and down scrolling” on page 15](#)
- [“Clearing the screen” on page 15](#)
- [“Deleting comment lines” on page 15](#)
- [“Inserting and deleting lines” on page 16](#)
- [“Displaying continuation lines of command output” on page 16](#)
- [“Delaying or holding automatic updating” on page 17](#)

Up and down scrolling

You can scroll up and down using PFKeys or INFO-line commands.

PF19 and PF20 are the default scrolling keys (on terminals with 24 programmed function keys). They are set to /UP and /DOWN (alias /U and /D), and scroll one physical screen at a time. You can change the default scroll amount with the OPTN command.

You can also scroll by using the /UP and /DOWN INFO-line commands. Both /UP and /DOWN accept the following arguments:

nnn

Scrolls *nnn* lines (*nnn* can be any number from 1–999).

PAGE

Scrolls a physical screen at a time.

MAX

Scrolls up or down the number of LROWS defined for your terminal. The short form of MAX is M.

CSR

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

As an alternative to /UP and /DOWN, you can use the INFO-line command /TOP to scroll to the top of the logical screen and /BOTTOM to scroll to the bottom of the logical screen.

You can also type any of the /UP and /DOWN arguments (such as *nnn*, MAX, or CSR) on the INFO-line before you press the PF key to change the scroll amount. If, for example, you type MAX on the INFO-line and then press PF20, OMEGAMON scrolls the maximum number of lines.

Clearing the screen

Use the clear screen immediate command to clear the screen of a menu, screen space, or any commands displayed on your screen to make space to enter commands.

The clear screen command consists of two periods followed by two blanks. Enter `. . bb` in the input area of any line following the INFO-line, and it clears all of the screen below it.

Deleting comment lines

If your screen is cluttered with lines of comment text, and you want to clear the comments but retain other command text, use the /DCL INFO-line command or the .DCL immediate command. /DCL clears all comment lines on the logical screen; .DCL clears all comment lines beneath its entry line.

The following display is an example of using /DCL to clear help text.

```
/DCL----- TSO      OM/DEX      V420./C SYS  02/12/09 10:32:22 1
>.SPT
>      Sets pattern number n to the value cccccc for generic selection.
.SPT/1 >M*      <
DEVP/1  MVSA21  MP220A  MP220D  MVSD21  MVSE21
>dadr
>      Device unit address (cuu)
>dio
>      Device I/O count (For rate use: 'DIO.R')
>dres
>      Number of outstanding RESERVEs from this CPU for this device
.DEFON  >> OB0130: Definition mode ENABLED. <<
DEVP/1
dadr
dalc
dio
.DEFOF  >> OB0130: Definition mode DISABLED. <<
```

On the next cycle, the following screen appears.

```

----- TSO      OM/DEX      V420./C SYS  02/12/09 10:32:22 1
.SPT/1 >M*      <
DEVP/1  MVSA21  MP220A  MP220D  MVSD21  MVSE21
.DEFON  >> 0B0130: Definition mode ENABLED. <<
DEVP/1
dadr
dalc
dio
.DEFOF  >> 0B0130: Definition mode DISABLED <<

```

Inserting and deleting lines

You can insert lines into a screen or delete lines from a screen using immediate commands.

To insert blank lines into a screen, use the following immediate command:

```
.I nn
```

The variable *nn* indicates the number of lines to insert. The default is 1. The *nn* argument must begin in column 6.

OMEGAMON inserts the new lines preceding the line where you type the insert command. Therefore, all other lines currently on the screen following the inserted line shift downward. When the command executes, the line you typed over with the insert command restores to its original data. For example, if you first display the following screen:

```

DISK    VMXA04  VMXA05  VMSP50  VMHP02  OMONVM  DOSTST  DP215R  DOSRES  +
dadr    1A0     1A1     1B0     1B1     2A7     2B0     4F1     4F2
dalc    8       16      6       12      24      16      32      8

```

and you enter `.I` on the second line, as in the following screen:

```

DISK    VMXA04  VMXA05  VMSP50  VMHP02  OMONVM  DOSTST  DP215R  DOSRES  +
.I      1A0     1A1     1B0     1B1     2A7     2B0     4F1     4F2
dalc    8       16      6       12      24      16      32      8

```

OMEGAMON inserts one blank line after the first line, and restores all the original data on the second line:

```

DISK    VMXA04  VMXA05  VMSP50  VMHP02  OMONVM  DOSTST  DP215R  DOSRES  +
dadr    1A0     1A1     1B0     1B1     2A7     2B0     4F1     4F2
dalc    8       16      6       12      24      16      32      8

```

To delete lines from a screen space, use the `.D nn` immediate command. The `.D nn` works the same way as `.I nn`. If you do not specify *nn*, the value defaults to one deletion line.

You can also delete a block of data from the physical screen. To do this, enter `.DD` on the first and last lines of the block.

Displaying continuation lines of command output

Some major commands select a series of items. Often, the display output does not fit onto one line. If this is the case, the major command displays only the first line of output, and a plus sign appears at the right

of the line to indicate that more data is available. You can see a count of the number of items that the command will select.

To display a count of the number of items that a major command will select, enter a pound sign (#) in column 1 before the command. The following example shows that there are 20 online disks for the DISK command to display:

```
#DISK    20
```

Continuing major command output

You can use a number of different continuation characters and commands to control the output display for major commands.

1. Enter a less-than sign (<) in column 1 in front of a major command to generate all output at once when you execute it. For example:

```
<DISK    VMXA04    VMXA05    VMSP50    VMHP02    OMONVM    DOSTST    DP215R    DOSRES +
          CPM023    DBRCPROD    DLSPROD    IRLMPROD    MPP01     MPP02
```

2. After you issue a major command for the first time, you can put a number in column 1 to specify how many lines to skip before the next display.
3. You can repeat a major command to display only the next line of output.

Continuing major and minor command output

If a major command lists a series of items that continues for more than one line, any minor command you enter after the major applies only to the last line of output. Therefore, if you enter a minor command after the third line of major command output, that minor command only generates information about the third line of items listed by the major. By default, the major and minor commands must be repeated once for each line of available output. The .RC command is a shortcut to this process.

.RC automatically repeats the major and the minor until all available lines of data are displayed. Enter the major command once, followed by the minor command, and then .RC beneath the minor. The following display is a sample of the output:

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

DISK    CPM023    DBRCPROD    DLSPROD    IRLMPROD    MPP01     MPP02
dadr    4F3      4F4      4F5      4F6      4F7      4F8
.RC
```

Delaying or holding automatic updating

The delay and hold features temporarily defer updating if you are in dedicated mode or in VTAM mode with automatic updating in effect (see the .AUP command).

While OMEGAMON is automatically updating, you can continue to enter commands. If OMEGAMON detects a cursor movement since the last update, it defers processing in order to avoid executing half-entered input. The words I/O Delay appear on the INFO-line, and the screen is not updated for the number of cycles specified with the IODELAY keyword of the .SET command.

If your screen display contains data that you want to study for longer than the normal automatic update interval, you can use the hold feature to temporarily freeze the screen image. To do so, home the cursor, then move it back one space to the blank space in column 1 of the INFO-line. The words Hold Mode appear on the INFO-line, and the information on the screen does not change until you move the cursor away from row 1, column 1.

Chapter 2. Configuring and running in different modes

You can run OMEGAMON for MVS in four different optional processing modes.

- [“Dedicated mode” on page 19](#)
- [“VTAM mode” on page 23](#)
- [“TSO mode” on page 27](#)
- [“ISPF mode” on page 29](#)
- [“Authorizing OMEGAMON libraries for TSO and ISPF modes” on page 33](#)

Dedicated mode

In dedicated mode, OMEGAMON for MVS can report hardware and software problems so severe that they disable other mechanisms, including z/OS system consoles. Dedicated mode offers the highest OMEGAMON availability. In this mode, OMEGAMON does not rely on VTAM. Instead, OMEGAMON communicates using the execute channel program (EXCP).

In dedicated mode, OMEGAMON refreshes the screen automatically every few seconds without operator intervention. The default refresh cycle is five seconds; however, you can change this interval to suit your needs.

Best practice: run page-fixed

In dedicated mode, OMEGAMON should be run page-fixed and nonswappable to provide higher availability for exception analysis. If OMEGAMON is page-fixed, it is protected from paging and swapping problems. If OMEGAMON is not page-fixed, one of the following conditions may be in effect:

- Swapped every cycle on a long wait, which increases swapping, and device and channel activity
- Vulnerable to problems with the ASM, page or swap data sets, devices, control units, and channels
- Vulnerable to unilateral or exchange swaps by the SRM

Issue the OMEGAMON IOPT command to display and customize page-fixing options. IOPT provides options for setting OMEGAMON storage page-fixed in memory, marking OMEGAMON non-swappable, and issuing DASD reserves when members are saved in RKOMPCSV. If you make changes to IOPT, be sure to save your profile using the PPRF or IPRF command. Then recycle the OMEGAMON task.

Customizing for dedicated mode

To configure OMEGAMON to run in non-VTAM dedicated mode, you must complete several customization steps and start a session in dedicated mode.

To configure OMEGAMON to run in non-VTAM dedicated mode, complete the following steps, then start OMEGAMON as described in [“Starting OMEGAMON in dedicated mode” on page 20](#):

1. Edit *rhilev.rte.RKANSAMU(KOMPROC)* and modify the parameters to suit your site. [Table 1 on page 20](#) explains the parameters and shows their defaults.
2. Save KOMPROC in your PROCLIB.
3. SMS-managed DASD users only: if SMS class is assigned to system-generated data set names, you must exclude jobname OMEGAMON system-generated data sets from SMS control.

Parameter	Value	Description	Default
COLS	<i>nnn</i>	Number of columns on OMEGAMON for MVS terminal.	80
FOLD	Y or N	Specifies whether or not to fold the password to upper case.	Y
FSCR	<i>cccccc</i>	First screen space that appears when you log on to OMEGAMON for MVS.	OMINITZZ
LROWS	<i>nnnn</i>	Number of logical rows for the output area (must be between ROWS and 9999).	99
MODE	<i>cc</i>	Mode of operation for OMEGAMON for MVS.	CN
PREFIX	<i>ccccccc</i>	High level qualifier used to install OMEGAMON for MVS.	none
PRTY	<i>nn,nn</i>	OMEGAMON for MVS dispatching priority. Your system specifications might override this value.	15,15
REG	<i>nnnnK</i>	Region size in K.	1536K
<i>rhilev.rte</i>	<i>ccccccc</i>	The high-level qualifier of the runtime libraries at your site.	YYYYYY
ROWS	<i>nn</i>	Number of physical rows on OMEGAMON for MVS terminal.	24
SYS	<i>cccc</i>	4-character system ID to display on the first row of OMEGAMON for MVS screen. If SYS=SMF, OMEGAMON for MVS uses the SMF system ID.	SMF
TIM	<i>nnnn</i>	OMEGAMON for MVS CPU time limit.	1439
UNIT	<i>uuuu</i>	Device address of OMEGAMON for MVS terminal. If you specify <i>uuuu</i> , OMEGAMON for MVS ignores the CONSOLE DD statement.	none
USR	<i>cc</i>	2-character profile suffix. Indicates which OMEGAMON user profile you are using.	none

Starting OMEGAMON in dedicated mode

To start OMEGAMON in dedicated mode, use the following procedure.

1. Make sure a dedicated 3270 is available.
2. If the terminal is a z/OS console, vary the device online:

```
V cuu,ONLINE
```

where *cuu* is the device number.

3. If the terminal is assigned to VTAM, enter the following command:

```
V NET,INACT,I,ID=termid
```

The variable *termid* is the name by which VTAM knows the terminal.

4. Enter the z/OS START command. (The operands may be different for your installation. They must match the name of a member in the PROCLIB concatenation.)

For example,

```
S KOMPROC,U=64
```

is a typical command to start OMEGAMON in dedicated mode. This example shows KOMPROC as the name of the started task and 064 as the device (UNIT) address.

To alter the standard version of OMEGAMON (change the number of PF keys, rows, columns, and other parameters), specify session parameters at OMEGAMON startup. See [“Customizing for dedicated mode” on page 19](#) for details.

You are now in OMEGAMON. The first screen that appears is the copyright screen. This copyright establishes the proprietary rights of IBM corporation to OMEGAMON.

Stopping OMEGAMON in dedicated mode

There are several methods for stopping OMEGAMON in dedicated mode.

You can stop a dedicated session in any of the following ways:

- Enter /STOP starting in column 2 of the first line of the screen:

```
/STOP_____ #01  DEDI  OM/DEX  V420./C A083 02/12/09  7:03:37  5B
EXSY      OMEGAMON/MVS Exception Analysis
+  ++++++
+  + Warning: CPU 0 not active  +
+  ++++++
+
```

- Issue the z/OS MODIFY command from the operator’s console:

```
F KOMPROC,P cuu
```

- To stop OMEGAMON and all dedicated sessions associated with that session, issue the z/OS STOP command from the operator’s console:

```
P KOMPROC
```

This action terminates the OMEGAMON address space.

Using secondary dedicated terminals

In dedicated mode, you can use a secondary terminal to display the output from a primary OMEGAMON session terminal. You cannot use the secondary terminal for input.

The .CN command starts and stops an OMEGAMON secondary terminal and accepts a unit address so that you can allocate the device interactively. You can also switch the functions of the primary and secondary terminals with .CN.

Running multiple dedicated sessions

You can run multiple dedicated OMEGAMON terminal sessions under one address space. After you bring up the first terminal normally, you can start and control additional sessions from within OMEGAMON or using z/OS operator commands.

To start the session from the operating system, use the z/OS MODIFY command with the following parameters:

```
F KOBROUTR,S CN,PROD=KOMMV510,UNIT=ccuu
```

where:

ccuu

Specifies the address of the secondary terminal. The PROD and UNIT parameters are required to start the session.

To start the session from within OMEGAMON, use the .MFY command:

```
.MFY S CN,PROD=KOMMV510,UNIT=ccuu
```

You can override default parameters for additional sessions by including them in the .MFY immediate command.

Issuing OMEGAMON commands from the system console

In dedicated mode, an operator can issue an OMEGAMON command from the system console using the z/OS MODIFY command.

The format for doing this is:

```
F KOMPROC,F cuu, cccccc
```

where *cuu* is the device address and *ccccc* is an OMEGAMON command. In the following example, the .LOGOUT command, which prints the current log, is sent to terminal 064.

```
F KOMPROC,F 064,b.LOGOUT
```

The space before the .LOGOUT command (b) is required because OMEGAMON commands generally start in column 2.

Only the command name may be used, no additional parameters. For example, the following command would not be allowed since IM* is an argument.

```
F KOMPROC,F 064, .SPT/1 IM*
```

Security in dedicated mode: the re-logon feature

Security in dedicated mode differs from the other modes, because, at startup time, there is no user ID or password associated with the session. Therefore, the only security available by default is internal security. In order to utilize external security on the dedicated session, you must use the re-logon feature.

The re-logon feature is a function of the /PWD command (see the *IBM OMEGAMON for z/OS: OMEGAMON for MVS Command Reference* for details on the /PWD command) that allows you to enter a user ID and password to the external security package from an active OMEGAMON session. This allows you to alter the security level of your session without stopping your session.

Type in the /PWD INFO-line command and your user ID as in the following example:

```
/PWD user01_____OMINITZZ DED . V500./C A083 01/01/99 17:03:37
```

Press Enter and type in your external security password at the prompt.

Note the following points regarding the use of the re-logon feature:

- Be sure not to mark the /PWD command as EXTERNAL=YES in the security table because, in dedicated mode, you must use /PWD to log on to external security.
- You can determine in your user exit what the default action should be when the user ID or logon password supplied is not valid. For example, you may specify the disabling of all OMEGAMON commands marked as EXTERNAL=YES, or you may specify that the session reverts to the previous user ID. The available options are explained in the sample exit routines.

- If you use the re-logout feature and your password has expired, you cannot enter a new one via the /PWD command. To re-establish the expired password, you must use the external security package procedures.

VTAM mode

In VTAM mode, OMEGAMON lets you run sessions from a VTAM terminal directly, without the intervention of an intermediate online application, such as TSO. Systems programmers or performance analysts can use VTAM mode to analyze MVS performance in real time when the extra availability of dedicated mode is not required, and when TSO mode is undesirable.

In VTAM mode, VTAM controls the 3270 terminal with which you access OMEGAMON. OMEGAMON VTAM support is provided through KOBVTAM, a VTAM application program. This application lets users log on to their own copy of OMEGAMON from any 327x terminal in the VTAM network. Of course, each session requires some additional virtual storage for buffers and work areas, but all sessions share a single copy of OMEGAMON. Consider this when you specify the REGION size on the KOBVTAM job step.

When you installed OMEGAMON for z/OS, the OMEGAMON for MVS VTAM mode started task was automatically installed as a member in your RKANSAMU library. The name of the member is the name you specified for the started task when you configured the product. Read the instructions in this topic if you want to set up an additional OMEGAMON for MVS VTAM mode system that can execute with different processing options, or if you want to change the options on the original OMEGAMON component.

Customizing the VTAM startup procedure

You must customize the startup procedure to start an OMEGAMON session in VTAM mode.

To customize for VTAM mode, complete the following steps.

Note: Because some items involve changes to VTAM system data sets, you may wish to consult your VTAM network systems programmer before you proceed.

1. Edit *rhilev.rte.RKANSAMU(cccccccc)* and modify the parameters to suit your site (where *cccccccc* is the name for the started task you specified when you configured OMEGAMON for MVS). [Table 2 on page 24](#) explains the parameters and shows their defaults.

You can specify most of the OMEGAMON keywords in either the PARM string in *cccccccc* or the VTAM logon DATA string. Exceptions are shown in the table. If you specify any parameter in both places, OMEGAMON uses the value in the VTAM logon DATA string.

The following list describes the hierarchy of parameters, starting with the lowest level:

- OMEGAMON comes with its own set of defaults.
 - For the ROWS parameter only, you can override the OMEGAMON defaults with values from the VTAM mode table entry.
 - PARM string values in *cccccccc* override OMEGAMON defaults and values from the VTAM mode table entry.
 - LOGON DATA string parameter values override any other values.
2. Make sure that the data set you specify on the RKOMPCSV DD statement is the same as the first data set you specify on the RKOMPROC DD statement.
 3. Make sure that the data set you specify on the RKOMPFSV DD statement is the same as the first data set you specify on the RKOMPROF DD statement.
 4. Save *cccccccc* in one of your system PROCLIBs, such as SYS1.PROCLIB.
 5. If you want the VTAM application ID to be available during the VTAM initialization process, define the application ID (APPLID) in SYS1.VTAMLST.

You can define the application ID either in a separate member in the SYS1.VTAMLST data set, or in the member that contains your existing application list. A separate member is more convenient for testing purposes, because you can vary the major node (member) for the application list inactive, modify it, and then vary it active again without disturbing other VTAM applications. However, if you store the

application ID in a separate member in SYS1.VTAMLST, the member name you use must be different from the application name and the ACBNAME (if it is not, the ACB does not open). The application name and the ACBNAME can be the same.

The following sample is found in *rhilev.rte.RKANSAM(KOBVTAPL)*:

```
VTAMAPPL VBUILD TYPE=APPL
OMVTAM APPL EAS=32,ACBNAME=OMVTAM
```

<i>Table 2. VTAM mode parameters</i>			
Parameter	Value	Description	Default
APPL	ccccccc	1 to 8 character application ID that opens the VTAM ACB. It must match the ACBNAME= parameter on the application ID in SYS1.VTAMLST. You can enter it only on the PARM string for the KOBVTAM program.	OMVTAM
AUP	YES or NO	Specifies whether the OMEGAMON sessions are in automatic update mode.	NO
DATA	YES or NO	Specifies whether OMEGAMON is to use a logon DATA string. If you specify NO, OMEGAMON uses a VTAM interpret table. You can enter it only on the PARM string for the KOBVTAM program.	YES
DC	Y or N	Specifies whether OMEGAMON will compress the 3270 data stream before it is sent to a terminal.	N
FOLD	Y or N	Specifies whether or not to fold the password to upper case.	Y
FSCR	ccccccc	Changes default for all logons to VTAM. You can enter it in either the PARM string for the KOBVTAM program or the logon DATA (must be 1 to 8 characters).	OMINITZZ
LROWS	nnnn	Number of logical rows for the output area. LROWS must be between rows and 9999, where rows is the number of physical lines on your screen. Candle recommends a minimum LROWS value of 99 to ensure the proper execution of screen spaces.	3000
OM	KOBROUTR	Specifies which OMEGAMON load module VTAM should attach.	KOBROUTR
PROD	KOMMVXXX	Product module attached by KOBROUTR.	KOMMVXXX
PRTCT	ccccccc	1 to 8 character password OMEGAMON uses to open the VTAM ACB. It must match the PRTCT= parameter in the VTAMLST APPL statement. You can enter it only on the PARM string for the KOBVTAM program.	no ACB password
PSWD	ccccccc	1 to 8 character password for users to log on to OMEGAMON in VTAM mode. If you specify it in the PARM string of the KOBVTAM program, you must also specify it in the DATA string of the VTAM logon command.	no logon password
RHILEV	ccccccc	The high-level qualifier of the runtime libraries.	

Table 2. VTAM mode parameters (continued)

Parameter	Value	Description	Default
SWAP	YES or NO	NO specifies that OMEGAMON for MVS will mark its address space non-swappable. (OMEGAMON for MVS must execute from an APF-authorized library to be non-swappable.) The address space will only be marked non-swappable while at least one OMEGAMON session is executing under it. When the last session terminates, the address space is marked swappable until a new session is started.	NO
TIMEOUT	0-999	Specifies the maximum number of minutes that a VTAM mode session can be idle. Once a timeout value is set, any VTAM mode session that is idle for the specified number of minutes is terminated by KOBVTAM. 0 indicates that an idle session will remain idle until the user terminates it. Note that any OMEGAMON user session initiated by VTM1 is also considered to be a VTAM mode session with respect to timeout processing.	0
UMAX	01-99	Maximum number of users who may log on. You can enter it only on the PARM string for the KOBVTAM program.	10
USR	cc	2-character profile suffix. Indicates which OMEGAMON user profile you are using.	/l

VTAM mode optional features

VTAM mode offers several option features you can invoke.

The following are optional features of VTAM mode:

- For automatic logons, add the LOGAPPL= parameter to an LU or TERMINAL statement in SYS1.VTAMLST.
- To start the OMEGAMON VTAM interface automatically at IPL, add a START command for the OMEGAMON VTAM interface started task procedure to SYS1.PARMLIB(COMMNDxx).
- To simplify the logon process, update your USS (Unformatted System Services) table or tables.

Important: If you use a logon interpret table, problems may occur because the entire logon command is passed in the logon message instead of just the upper and lowercase letters so that the interpret table does not recognize the logon command. For example, use LogoN instead of LOGON or logon.

- To control who may log on to the OMEGAMON VTAM interface, use the PSWD= parameter or a VTAM session authorization exit routine. A routine might be more effective because it controls which secondary logical units may log on to the OMEGAMON VTAM application interface. To create or update a VTAM authorization exit (ISTAUCAT), refer to the appropriate VTAM systems programming or installation reference.

Your site can also provide a logon security exit routine, which requires a user to supply a password known to RACF®, CA-ACF2, or CA-TOP SECRET. For more information, see [Configuring](#).

Starting OMEGAMON in VTAM mode

To start OMEGAMON in VTAM mode, use the following procedure.

1. Activate the application that starts OMEGAMON in VTAM mode from the console.

The format for starting this application is:

```
V NET,ACT,ID=cccccccc
```

where *cccccccc* is the application ID. For example:

```
V NET,ACT,ID=OMVTAM
```

2. To start OMEGAMON, enter:

```
S cccccccc
```

where *cccccccc* is the started task name.

If you try to start OMEGAMON and the VTAM application ID is not active, the KOBVTAM program waits up to 30 minutes for the application ID (OMVTAM) to be varied active before it terminates.

3. After OMEGAMON starts, you can activate secondary logical units (SLUs) to be used as OMEGAMON terminals.

The format for starting an SLU is:

```
V NET,ACT,ID=nodename
```

where *nodename* is the VTAM node name. For example:

```
V NET,ACT,ID=L064
```

You can start an OMEGAMON session in one of the following three ways:

- From a primary or secondary terminal
- From the network operator's console
- From the definition of the terminal itself
- From a primary or secondary terminal, start the logon with the following command:

```
LOGON APPLID(cccccccc) DATA('.....')
```

The DATA parameter has two uses:

- To specify parameters OMEGAMON uses. This example specifies the password to OMEGAMON:

```
LOGON APPLID(cccccccc) DATA('PSWD=MYSECRET')
```

If this password matches the password supplied by the installation in the OMEGAMON startup parameters, the logon to OMEGAMON proceeds; if not, it is rejected.

- To specify parameters that VTAM passes to OMEGAMON. This example specifies to OMEGAMON that a logical screen size of 80 rows is being used:

```
LOGON APPLID(cccccccc) DATA('LROWS=80')
```

Specify ROWS and COLS for a VTAM terminal only if that information is not available (or is not correct) in the VTAM logon mode table for the terminal.

You can mix parameters specific to OMEGAMON and to VTAM in the parameter string you supply via the DATA parameter of the VTAM LOGON command. OMEGAMON requires single quotes, as the examples indicate.

Installers may have modified the LOGON syntax with the unformatted system services (USS) table.

To terminate this display, enter LOGOFF in the update area and press Enter.

- From the network operator's console, start the logon with the following command (from a z/OS console or an NCCF terminal):

```
V NET, ID=luname, LOGON=cccccccc
```

where *luname* is the secondary lu or terminal.

- From the definition of the terminal itself, start the logon by adding the LOGAPPL parameter to the LOCAL or LU definition in SYS1.VTAMLST:

```
luname LOCAL . . . , LOGAPPL=cccccccc, . . .
```

where *luname* is the secondary lu or terminal.

Use VTAM procedures to recover from error conditions in other VTAM applications. To recover a session, vary the secondary lu inactive and then active again. If there are serious problems, you can terminate all sessions by varying the primary lu inactive, or terminate all sessions and the KOBVTAM address space by using the OS STOP, CANCEL, or FORCE commands.

Stopping OMEGAMON in VTAM mode

To stop an OMEGAMON session running under VTAM, enter /STOP at the beginning of the top line of the screen, as shown in [Figure 9 on page 29](#). Only the individual OMEGAMON session terminates. The VTAM application (OMVTAM) continues to run even when no sessions are active.

To stop the OMVTAM application, issue this z/OS STOP command from the operator's console:

```
P cccccccc
```

where *cccccccc* is the started task name.

If any sessions are active, the CLOSE process terminates them.

If the VTAM application ID is changed to inactive while OMEGAMON runs in VTAM mode, the OMVTAM application terminates.

TSO mode

In TSO mode, OMEGAMON runs as a normal TSO command on any of its supported terminals. OMEGAMON users who do not require the high availability of dedicated mode can use TSO mode to access OMEGAMON. In TSO mode, the screen does not refresh automatically; you must press Enter.

Before you can begin to customize for TSO mode, be sure that you have built and loaded a runtime environment.

If you are using RACF external security, OMEGAMON must be APF-authorized to accommodate RACHECK processing (see [“Authorizing OMEGAMON libraries for TSO and ISPF modes” on page 33](#)).

Customizing for TSO mode

To customize for TSO mode, complete the following steps:

1. Edit the startup CLIST *rhilev.rte.RKANSAMU(KOMCLIST)*.

If your site runs TSO/E, use member KOMCLSTE instead. KOMCLIST and KOMCLSTE are correct for both FB and VB data sets. For FB data sets, use this member as is. For VB data sets, copy it into a VB data set and renumber it. The CLIST uses columns 9 through 72. FB and VB numbering do not affect its contents.

Modify the parameters in KOMCLIST to suit your site. [Table 3 on page 28](#) explains the parameters and shows their defaults.

2. Save KOMCLIST in one of your CLIST libraries.
3. Make sure your TSO region size is at least 1.5 M.

Table 3. TSO mode parameters

Value	Parameter	Description	Default
COLS	<i>nnn</i>	Number of columns on the OMEGAMON terminal.	Terminal size
DEBUG	Indicates debug mode.	If you specify this parameter, you can watch the execution of each line of the CLIST.	Not specified
FOLD	Y or N	Specifies whether or not to fold the password to upper case.	Y
FSCR	<i>ccccccc</i>	First screen space that appears when you log on to OMEGAMON.	OMINITZZ
LROWS	<i>nnnn</i>	Number of logical rows for the output area. LROWS must be between rows and 9999, where rows is the number of physical lines on your screen.	3000
MODE	<i>cc</i>	Mode of operation for OMEGAMON. TS for TSO mode.	TS
PROD	KOMMVXXX	Product module attached by KOBROUTR.	KOMMVXXX
QUICK		OMEGAMON files are not allocated.	Not specified
RHILEV	<i>cccccncc</i>	High-level qualifier for the runtime libraries.	YYYYYY
ROWS		Number of physical rows on the OMEGAMON terminal.	Terminal size
SYS	<i>cccc</i>	4-character system ID to display on first row of the OMEGAMON screen. If SYS=SMF, OMEGAMON uses the SMF system ID.	SMF
USR	<i>cc</i>	2-character profile suffix. Indicates which OMEGAMON user profile you are using.	/l

Starting OMEGAMON in TSO mode

To start OMEGAMON in TSO mode, use the following procedure:

- From the TSO READY prompt or from the TSO command processor option under ISPF (usually option 6), run the OMEGAMON CLIST:

```
%KOMCLIST
```

The CLIST name may be different for your installation. It must match the name of the member in the library where the CLIST resides, which must either be in your SYSPROC data set or concatenated to SYSPROC.

To alter the standard version of OMEGAMON, specify session parameters when starting OMEGAMON. You can set defaults in the KOMCLIST. Set the LROWS= default, which specifies the number of logical rows for the output area. See [“Customizing the VTAM startup procedure”](#) on page 23 for details.

You are now in OMEGAMON. The first screen that appears is the copyright screen. This copyright establishes the proprietary rights of IBM to OMEGAMON.

- To continue, press Enter.

Stopping OMEGAMON in TSO mode

To stop an OMEGAMON session in TSO mode, enter /STOP at the beginning of the first line of the screen, as shown in [Figure 9 on page 29](#).

```
/STOP_____ #01   TSO   OM/DEX   V420./C A083 02/12/09   7:03:37 5B
```

Figure 9. Stopping an OMEGAMON for MVS session

Modifying virtual terminal pool definitions

If you use TSO mode and want to modify virtual terminal pool definitions, and your runtime environment does not share libraries with other runtime environments or with SMP/E, take the following steps. For more information about sharing and non-sharing runtime environments, see [Configuring](#).

1. Review comments in the following product members in the RKANSAMU data set to ensure you understand the members' contents and purpose:

- KOBVTPL
- KOBVTPLX
- KOBVT1AP

2. Define your virtual terminals and logmode names to the VTM1 program by updating member KOBVTPL in the RKANSAMU data set.
3. Assemble and link edit the KOBVTPL source to create the KOBVTPL load module using the JCL in member KOBVTPLX in the RKANSAMU data set. Load module KOBVTPL is stored in the RKANMODU data set.
4. If you have modified the terminal definitions (either the number of terminals or their names):
 - Update the VTAM node list member KOBVT1AP in the RKANSAMU data set.
 - Update your VTAMLST controls accordingly.

ISPF mode

In ISPF mode, IBM OMEGAMON for z/OS behaves differently according to the APAR level.

After APAR OA59414

In ISPF mode, you can run OMEGAMON for z/OS as a standalone application. You can use all of the OMEGAMON for z/OS functions, but you cannot use ISPF functions; for example, you cannot swap screens to other ISPF screens.

Before APAR OA59414

In ISPF mode, OMEGAMON for z/OS supports split-screen mode, so that you can switch between OMEGAMON for z/OS and other ISPF applications.

If you are using RACF® external security, OMEGAMON for z/OS must be APF-authorized to accommodate RACHECK processing (see [“Authorizing OMEGAMON libraries for TSO and ISPF modes” on page 33](#)).

OMEGAMON ISPF mode components

OMEGAMON includes a number of components used for ISPF mode.

The following list describes the components that OMEGAMON includes for ISPF mode:

- Load modules

KOBSPF

The main driver program that links to OMEGAMON.

KOBSPFAU

Used for program authorization.

KOBSPFSW

Also used for program authorization.

• CLIST

KOMSPF

Establishes OMEGAMON as an ISPF application and displays the KOMSPF screen.

KOMSPFU

Allocates the needed files and loads OMEGAMON.

• Panels

KOMSPF01

The main display panel on which OMEGAMON appears.

KOMSPF02

The KOMSPF invocation menu.

KOMSPF03

Called from the KOMSPF invocation menu, the panel defines the function keys for splitting and swapping screens.

KOMSPF04

The help screen for KOMSPF02.

KOMISR@P

ISPF Primary Option Menu to invoke OMEGAMON.

Configuring IBM OMEGAMON for z/OS to run in ISPF mode

How you configure OMEGAMON for z/OS to run in ISPF mode depends on the APAR level.

See:

[“After APAR OA59414” on page 31](#)

[“Before APAR OA59414” on page 31](#)

After APAR OA59414

To enable OMEGAMON for z/OS to run standalone in ISPF mode, complete the following steps.

1. Create a CLIST member that is called KOMSPFSX, and then copy your installation's KOMCLSTE or KOMCLIST CLIST member to it.

If you have not previously customized these CLISTS, sample CLISTS are created during runtime environment configuration and stored in the *&rhilev.&rte*.RKANSAMU data set.

2. Create a CLIST member that is called KOMSPFSI.
3. Delete the TSOLIB ACTIVATE DATASET command from the KOMSPFSX CLIST, and then add that command to the KOMSPFSI member.
4. Create a CLIST member that is called KOMSPFSC.
5. Delete the TSOLIB DEACTIVATE command from the KOMSPFSX CLIST, and then add that command to the KOMSPFSC member.
6. Copy the three new CLIST members (KOMSPFSX, KOMSPFSI, and KOMSPFSC) to a CLIST data set that is in your installation's SYSPROC DD concatenation.

Before APAR OA59414

To run OMEGAMON for z/OS in ISPF mode, complete the following steps. Make sure your TSO region is at least 1.5 M.

1. Copy the KOMSPF and KOMSPFU CLISTs from *rhilev.midlev.RKANSAMU* to the CLIST library and include them in the SYSPROC concatenation.
2. Edit and update KOMSPFU with the high-level qualifier you are using for OMEGAMON for z/OS data sets.
3. If you want to access an unauthorized copy of OMEGAMON for z/OS, set the OMDSNAU variable to a single blank.
4. Copy the following panels from *thilev.RKANISP* to the ISPF panel library: KOMSPF01, KOMSPF02, KOMSPF03, and KOMSPF04.
5. (Optional) Using *rhilev.RKANISP(KOMISR@P)* as a model, modify the ISPF Primary Option Menu to invoke OMEGAMON for z/OS.

Starting IBM OMEGAMON for z/OS in ISPF mode

How you start OMEGAMON for z/OS in ISPF mode depends on the APAR level.

See:

[“After APAR OA59414” on page 31](#)

[“Before APAR OA59414” on page 31](#)

After APAR OA59414

To run OMEGAMON for z/OS standalone in ISPF mode, use the following procedure:

1. Perform the steps that are in [“Configuring IBM OMEGAMON for z/OS to run in ISPF mode” on page 30](#).
2. At the TSO READY prompt, enter the following command:

```
KOMSPFSI
```

You can put the invocation of KOMSPFSI in a TSO session initialization CLIST.

3. Enter ISPF, and then enter the following command on an ISPF option or command line:

```
TSOKOMSPFSX
```

A OMEGAMON for z/OS Classic ISPF standalone user session starts.

4. After you exit from ISPF, you can deactivate the OMEGAMON for z/OS TSOLIBs by invoking the KOMSPFSC CLIST from a TSO READY prompt.

Before APAR OA59414

To start OMEGAMON for z/OS in ISPF mode, use the following procedure.

1. From the TSO command processor option (usually option **6**) of the ISPF menu, execute the OMEGAMON for z/OS CLIST KOMSPF prefixed with % (%KOMSPF).

The CLIST name may be different for your installation. It must match the name of the member in the library where the CLIST resides, which must either be in your SYSPROC data set or concatenated to SYSPROC. The following screen appears after you enter %KOMSPF.

```

----- OMSPF - INVOCATION MENU -----
OPTION  ==> _
          1  ALLOCATE      - ALLOCATE OMEGAMON DATASETS
          2  ASSIGN KEYS  - ASSIGN SPLIT-SWAP PFK
          3  BEGIN        - BEGIN PRODUCT EXECUTION
          X  EXIT          - EXIT THIS SCREEN

          FOR OPTION 3,  SPECIFY:

          USER SUFFIX ==> nn
          LROWS       ==> nn   (24 - 999)
          FIRST SCREEN==>

```

Figure 10. Invocation Menu displayed by KOMSPF CLIST

2. Select option **1** to allocate the needed files.

You must allocate data sets the first time you request OMEGAMON for z/OS during each program session.

3. Select option **2** to reassign the PF keys to OMEGAMON for z/OS.

If you want to remove the ISPF split and swap functions, set the PF key numbers to 0.

4. If you want to override the installation-set program parameters, specify the following:

USER SUFFIX

The user profile (/C is the default, and specifies the default profile).

LROWS

The number of rows in your logical screen.

FIRST SCREEN

The screen space that appears first at OMEGAMON for z/OS startup. In this screen space, you can include, for example, commands to turn on extended color capabilities or to set your report parameters.

Note: You can request both basic and extended color options in ISPF mode if your terminal supports them. However, ISPF does not support the following extended color support highlighting features: blinking, reverse video, and high intensity.

5. Select option **3** to begin OMEGAMON for z/OS.

You are now in OMEGAMON for z/OS. The first screen that appears is the copyright screen. This copyright establishes the proprietary rights of IBM to IBM OMEGAMON for z/OS. To continue, press Enter.

Stopping OMEGAMON in ISPF mode

When you stop OMEGAMON under ISPF (/STOP), you return to the OMSPF invocation menu. From this menu, select option **X** to return to ISPF.

Modifying virtual terminal pool definitions

If you use ISPF mode and want to modify virtual terminal pool definitions, and your runtime environment does not share libraries with other runtime environments or with SMP/E, take the following steps:

1. Review comments in the following product members in data set RKANSAMU to ensure you understand the members' contents and purpose: KOBVTPL, KOBVTPLX, KOBVT1AP.
2. Define your virtual terminals and logmode names to the VTM1 program by updating member KOBVTPL in the RKANSAMU data set.
3. Assemble and link edit the KOBVTPL source to create the KOBVTPL load module using the JCL in member KOBVTPLX in the RKANSAMU data set. Load module KOBVTPL is stored in the RKANMODU data set.
4. If you have modified the terminal definitions (either the number of terminals or their names):

- Update the VTAM node list member KOBVT1AP in the RKANSAMU data set.
- Update your VTAMLST controls accordingly.

Authorizing OMEGAMON libraries for TSO and ISPF modes

The load library you specify in the CALL statement of OMEGAMON for MVS must be APF-authorized or part of the linklist concatenation.

As a rule, any program CALLED via TSO is not APF-authorized, regardless of what library it comes from or its authorization code. To allow OMEGAMON for MVS to retain APF-authorization when you invoke it, you must put KOBROUTR and KOBSPFAU in the TSO authorized call tables CSECT IKJEFT8.

```

ENTRY APFPTABL
IKJEFT8 CSECT
          DC    CL8'IKJEFT8'
          DC    CL8' 80.240'                Assembly Date
APFPTABL DC    CL8'IEBCOPY'                This line may already exist
          DC    CL8'KOBROUTR'
          DC    CL8'KOBSPFAU'
          DC    CL8' '                      Null entry (end-of-table)
          END

```

Figure 11. Sample CSECT to add OMEGAMON to TSO authorized list

You can APF-authorize KOBROUTR and KOBSPFAU by listing them in the AUTHPGM NAME(...) parameter of member IKJTSO00 in SYS1.PARMLIB, instead of using IKJTABLS.

Important: Do not remove the last 8-byte entry, which marks the end of the table.

In *rhilev.rte.RKANSAMU(KOMSPFU)*, insert the name of your installation's authorized library that contains OMEGAMON for MVS load modules, as in the following example:

```
SET &OMLIB = &STR(rhilev.rte.RKANMOD)
```

All of the usual restrictions on authorized programs that you run under TSO apply. In particular, if you issue the CALL command as a subcommand of another (unauthorized) command such as ISPF or IPCS, OMEGAMON for MVS is not authorized.

Chapter 3. Operational commands

The following topics describe some INFO-line and immediate commands that control OMEGAMON, as well as some generic operands.

Other operational commands that can be used in customizing your OMEGAMON profile are described in Chapter 5, “The User Profile Facility,” on page 77.

Command modifiers and help symbols

Some command modifiers can be used generically 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 may 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 to 35) that specifies the number of lines to skip from the last line displayed. For example, entering 3DISK on the first row of output from the DISK major command displays the 4th 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 generates this character in front of some commands after they execute to prevent them from being executed again on succeeding cycles. It is also displayed in front of help text. You can use it when creating 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 (+) appears 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 disappears from the screen on the next cycle.

Rate, difference, and percentage arguments

For any minor OMEGAMON command that normally displays a numeric value, you can add an argument that calculates the rate of a system event during an OMEGAMON interval or the difference from one interval to the next. Statistics that show rates or differences are often more useful than 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 CPU time, the rate calculated represents a scaled percentage of utilization (expressed as a decimal value) where the real time of the interval equals 100%.

OMEGAMON requires data from 2 cycles to calculate a rate or a difference. On the first cycle (the initialization cycle), a row of 8 periods (.) appears. The rate appears on the next cycle.

Be aware that 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 may select different busy disks at each screen update. When a rate argument is used with a minor of DSKB, the periods may appear after each update, indicating that OMEGAMON is initializing each time.
- If a command displays a value that has been reset to zero by the system between OMEGAMON intervals.

You can also use an argument that gives you a real percentage.

The following arguments must be entered in columns 6 and 7 after a minor command that displays a numeric value:

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

where

.D

Causes OMEGAMON to calculate the difference between the two most recent values of an event monitored by the specified command (cccc).

.R

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

.H

Displays a rate per hour.

.M

Displays a rate per minute.

.%

Displays a percentage.

Operational INFO-line and immediate commands

Some INFO-line and immediate commands control OMEGAMON.

The commands appear in alphabetical order, starting with special characters.

..bb

Clears the entire display down to the end of the logical screen (b indicates a blank).

Type:

Immediate

====

Draws a separator line across the screen.

Type:

Immediate

Format:

c====aa

c

For terminals that support an extended data stream, specifies 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 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, this command

```
R=====+
```

creates a line of red pluses across the screen.

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

====

Draws a separator line across the screen.

Type:

Immediate

/ATTN

Emulates the PA1 (program attention) key.

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

Type:

INFO-line

/AUP

Controls automatic update mode.

Type:

INFO-line

Format:

```
/AUP {ON|OFF}
```

ON

Enables automatic update mode.

OFF

Disables automatic update mode.

This command is valid only in VTAM mode. Automatic update mode is similar to running OMEGAMON 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 is automatically updating in VTAM mode, you can continue to enter commands. OMEGAMON delays 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 VTAM mode session).

.AUP is the equivalent immediate command.

.AUP

Controls automatic update mode.

Type:

Immediate

See /AUP, the equivalent INFO-line command, for the command description.

If .AUP is entered without an argument, OMEGAMON displays whether automatic update mode is ON or OFF.

.CN

Controls the specified secondary console in dedicated mode.

Type:

Immediate

Format:

.CNxxx

The variable xxx is the hexadecimal address of the secondary console.

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

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

.CN OP

Allocate (open) a secondary console for OMEGAMON 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.

Note: While the .CN command handles multiple terminals connected to a single session, the immediate command, .MFY, sets up multiple independent sessions.

.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 5 lines on the physical screen:

```
.D 5
```

/DCL

Deletes all comment lines on the screen.

Type:

INFO-line

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

.DCL

Deletes all comment lines following 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 disappears after it executes.

.DDb

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:

```
.DDb
DISK  SYSB24  TS0021  SYSB21  MVSA21
DSKB  MVSA21  PROD05  SYSA24
.DDb 150 334   D8B
```

/DEF

Inhibits automatic updating to allow screen space definition.

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 `/AUP` 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 canceled 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`).

This command allows you to define a screen space to include commands that comment themselves out or otherwise change form after execution..`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 displays the current definition mode status (`ON`, `OFF`, or `HOLD`).

DELT

Deletes a screen space from main storage or from the user's screen space library, `RKOMPCSV`, which is referenced by the `RKOMPCSV DD` statement.

Type:

Immediate

Format:

`DELTc aaaaaaaaaa where`

c

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

B or b

Deletes from both main storage and `RKOMPCSV`

I

Deletes from main storage (in-storage) only.

D

Deletes from RKOMPCSV only.

aaaaaaaa

The screen space name you want to delete. Specify the name starting in column 8.

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

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

```
DELT 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.

/DOWN

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

Type:

INFO-line

Format:

```
/DOWN cccc
```

cccc can be:

nnn

Scrolls *nnn* lines (from 1 to 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

Scrolls down so that the current cursor position is at the bottom of the physical screen. This is the default.

/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 will interpret 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 labeled `current` in the `.DSE` display. The entry pointer indicates which screen space in the stack has most recently been referenced with a `/STK INFO`-line command. If you issue the `/STK` command with an up or down argument, the pointer moves to the entry over or beneath 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       OMINITZZ  17987         11:56:00
+            Total stack size:  53961
```

.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*–*c2*). A character string can be 1–4 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 has 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 starting 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

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 very quickly, bypassing the screen display and the normal `OMEGAMON` interval. It is particularly useful in exception analysis for implementing the Automatic Screen Facility (ASF) or the Timed Screen Facility (TSF).

`.FGO` has a conditional screen fetch feature that fetches a screen space only if a condition is true.

Format:

```
n.FGO cccccccc
[CPSE  {=|EQ|GE|GT|LE|LT|NE} argument]
[MODE  {=|EQ|GE|GT|LE|LT|NE} argument]
[PREFIX {=|EQ|GE|GT|LE|LT|NE} argument]
[SMFID {=|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]
```

where

n

Is an optional numeric label that allows you to delay the fetch of screen space `ccccccc` 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 `ccccccc` is fetched on the next cycle.

ccccccc

Specifies the name of a screen space.

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

CPSER

CPU serial number. In the case of a multi-processor, this will compare the supplied CPU serial number with the serial numbers of all CPUs in the complex. If the relational argument is equal (= or EQ), OMEGAMON will fetch the screen space the first time it finds a match. If the relational argument is NE, OMEGAMON will fetch the screen space only after it has checked all of the CPUs in the complex.

MODE

The 3-character code for the OMEGAMON mode of operation. It is displayed on the INFO-line during a session. For a list, see [“INFO-line format” on page 12](#).

PREFIX

The OMEGAMON product code (for example, OM).

SMFID

The system ID from the SYS= startup parameter or the default SMF ID.

UNIT

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

USER

The user profile suffix from the USER= startup parameter.

&var

Allows you to set any comparison you want. The keyword and argument can be any variable name set with the .VAR command or any OMEGAMON-defined variable. The .VAR command lists OMEGAMON-defined variables.

The relational operators require blanks on either side except for 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 1- to 8-character value to which OMEGAMON compares the keyword. The keyword and argument can be any variable name set with the .VAR command or any OMEGAMON-defined variable. The .VAR command lists OMEGAMON-defined variables.

To protect against the possibility of a looping condition caused by the .FGO command, OMEGAMON 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 IBM-supplied profile). After the limit is reached, .FGO acts like .SGO (Screen Go) so that screen spaces continue to execute, but now they display on each cycle.

Therefore, if there was a loop caused by .FGO screen spaces, you will need to correct the condition and re-enable .FGO with the FGLOOP keyword of the .SET command.

Because FGLOOP=ON causes .FGO to display executing screen spaces, you may want to turn it on yourself to test your screen space fetch routines.

If multiple .FGO commands appear on one screen, the last .FGO command without a condition, or for which the condition is true, executes.

For example, to fetch screen space SAMPLE only if the terminal address is 05E1, enter:

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

or

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

See also the .SGO command.

/HELP

Describes HELP facilities (alias is /H).

Type:

INFO-line

The help screen space tells you how to find out more about the functions, features, and operation of OMEGAMON.

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 commands.

Type:

INFO-line

Format:

/HOLD ccc where

ON

Suspends OMEGAMON command execution.

OFF

Returns to normal OMEGAMON command execution.

/HOLD ON allows you to stop automatic updating 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 5 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 .ILC 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 REPORT log or the XLFLOG to the printer.

Type:

INFO-line

.LOG 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 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 execute .LOGPOP. This capability allows you to manipulate the log's status 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 will appear on the same line as the command.

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

or

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

POP

Restores the log to the status in effect when you executed the last .LOGPUSH. One of the following messages will appear 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.

/LOG is the equivalent INFO-line command. However, it 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 ccccccc ccccccc || ccccccc
```

The variables ccccccc 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 to load screen spaces ZZ1, ZZ2, and ZZ3 from the screen space library RKOMPROC to main storage.

```
LSCR ZZ1 ZZ2 ZZ3
```

.MFY

Simulates the result of a z/OS MODIFY command.

OMEGAMON simulates the receipt of a z/OS MODIFY command. Therefore, the target is the current OMEGAMON address space. .MFY allows you to start multiple OMEGAMON sessions in a single address space. The .MFY command is valid only in dedicated mode.

Type:

Immediate

Format:

```
.MFY S CN,UNIT=ccc,PROD=KOMMV510
```

UNIT=

Specifies the device address.

PROD=

Specifies the name of the OMEGAMON product module. Only one value is supported: KOMMV510.

For example, the next command shows that an additional dedicated session is to be started at address 577.

```
.MFY S CN,UNIT=577,PROD=KOMMV510
```

The PROD and UNIT parameters are required to start the session; the command also accepts the other parameters associated with starting a dedicated OMEGAMON session.

Note: While .MFY sets up multiple independent sessions, the .CN command handles multiple terminals connected to a single session.

.MIN

Lists all minor commands for the preceding major command.

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

Type:

Immediate

Format:

```
[H].MIN [nn|c1 c2]
```

(blank)

Without operands, .MIN displays all minor commands of the major.

H

The optional label H displays one-line help information for each of the minor commands. The following arguments allow you to limit the help display and avoid scrolling down to see the desired 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 to 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 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 2-character group names.

cc

Command Group

b

all immediate and major commands

AS

address space commands

- CN**
console commands
- DS**
disk commands
- DU**
dump commands
- DV**
device commands
- DX**
DEXAN commands
- EX**
exception analysis commands
- ME**
memory commands
- PA**
paging commands
- PC**
physical channel commands
- SA**
swap data set commands
- ST**
resource measurement facility (RMF) data
- SY**
system information commands
- TA**
tape 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 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 detects a program check, these names and addresses are useful to IBM Customer Support.

.NXE

Controls display of exceptions.

Type:
Immediate

Format:
.NXEccc

ON
Displays exceptions. This is the default.

OFF
Does not display exceptions.

Enter the .NXE immediate command without an argument to show the current status of the exceptions display.

The frequency for executing an exception 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 cycle, the .NXE command controls whether the EXSY command continues to display tripped exceptions on the cycles when they are not due for execution.

/O

Prints the existing OMEGAMON 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 /LOG or .LOG for complete information about this command.

.PCS

Displays OMEGAMON program check statistics.

Type:
Immediate

If OMEGAMON detects a program check, the information is useful to IBM Customer 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, .PFK displays all current PF key settings for command mode. PF keys without assignments do not appear on the screen. The default PF key definitions in command mode differ from those in the menu system.

c

Label E for redefining several PF keys at one time.

nn

PF key number.

aaaaaaaa

Screen space name (1 to 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 an 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
```

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 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 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 following the equal sign.
3. Press Enter.

The assignments you have entered will be in effect for the duration of the session.

Important: At startup, OMEGAMON executes the screen spaces containing default PF key assignments. The PF keys that you define with .PFK are in effect only for that OMEGAMON session. To make these assignments permanent, you must change the defaults in the screen spaces. For command mode, these screen spaces are @ZPFKDEF (PF keys 1–12) and @ZPFKDF2 (PF keys 13–24). For menu mode, they are @ZPFKNEW (PF keys 1–12) and @ZPFK2 (PF keys 13–24).

/PRINT

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

Type:

INFO-line

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

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

.PRM

Displays current values of the OMEGAMON startup command parameters.

Type:

Immediate

This command displays applicable OMEGAMON startup parameters in the following order:

IOMODE=cc

Current[®] OMEGAMON I/O mode. This is the 2-character code entered as the mode in the startup parameters.

SYS=cccc

Current OMEGAMON system ID. This is the same system ID that appears on the INFO-line.

USER=cc

User profile suffix. This is the same suffix that appears 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 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, .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 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 appear 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 will be cleared.

- The /PWD command can be entered with your user ID to perform the following functions.
 - Log onto an existing OMEGAMON 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 onto an existing session, you cannot change or update your password.

For details about OMEGAMON security features, see the *IBM OMEGAMON for 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 had repeatedly entered 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, .Rnnn 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 line commands (.D, .I) in that if you enter it above a line of data, that line is shifted down. Therefore, you do not have to insert a blank line in order 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 a large number of items, such as disks, only one line of output displays. The .RC command automatically repeats a major command and any minor or minors supplied until all available lines of output are displayed. Enter .RC after the major and any desired minors:

```
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:

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								

RENAM

Renames a screen space.

RENAM renames screen spaces in main storage or in the user-defined screen space library pointed to by ddname RKOMPCSV. It does not rename IBM-supplied screen spaces in the library pointed to by ddname RKOMPROC.

Type:

Immediate

Format:

RENAMc oldname newname where

c

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

B or b

Rename in both main storage and RKOMPCSV. This is the default.

D

Rename in RKOMPCSV only.

I

Rename in main storage (in-storage) only.

/REP

Replaces the existing saved screen space of the same name.

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

Type:

INFO-line

Format:

/REP ccccccc ,a where

ccccccc

Specifies the screen space name (1 to 8 characters)

a

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

B

Replaces in both main storage and RKOMPCSV. This is the default.

D

Replaces in RKOMPCSV 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 RKOMPCSV.

```
/REP SAMPLE,B
```

.REP

Displays and sets printer characteristics for the REPORT file.

Type:

Immediate

When you enter .REP, a series of keywords appears to let you set printer characteristics. If you are changing more parameters than will fit on one line, use the OUTP command instead. The keywords for .REP (and .XLG) duplicate the minors of the OUTP major.

When you change any of the parameters (except FOLD) and press Enter, OMEGAMON 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 will also appear if their default values have been modified previously. If not, you can type them in and define a new value.

ID1=

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

OMEGAMON generates separator pages and page headers with the appropriate job name printed in block letters on the pages. This is the default.

NONE

OMEGAMON does not generate separator pages and page headers.

cccccccc

OMEGAMON generates separator pages and page headers with ccccccc printed in block letters on the pages. ccccccc is up to 8 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 ddnames.

If you change any parameter other than HOLD=, OMEGAMON 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.

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

/RESHOW

Redisplays 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 have made 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.

.RTN

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

The .RTN command is required at the bottom 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 ASF and TSF described in [Appendix A, "Automating and logging features,"](#) on page 309.

Type:

Immediate

Format:

n.RTNcc aaaaaaaaa where

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 to 9 or the letters A to Z (representing 10 to 35). Each time the screen updates, *n* decrements by 1. When *n*=0, the current screen executes and OMEGAMON fetches the next screen space.

For example, the following command returns to the calling screen space after 7 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 ccccccc ,a` where

ccccccc

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

a

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

B

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

D or b

Saves the screen space to RKOMPCSV only.

I

Saves the screen space to main storage only.

Note: Because screen spaces are saved to the library pointed to by `ddname RKOMPCSV`, the `/SAVE` command works only if RKOMPCSV was created and properly concatenated at installation. If you are unable to save your screen space, or if your screen space was successfully saved but OMEGAMON is not accessing it, check with your installer.

This next example saves the current screen space `SAMPLE` in both main storage and RKOMPCSV:

```
/SAVE SAMPLE,B
```

For guidelines on creating screen spaces, see [Chapter 4, "Screen spaces," on page 71](#).

SCRN

Lists screen space member names.

Type:

Immediate

Format:

`SCRNc aa bb` where

c

Is one of the following arguments entered 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 beginning with characters *aa* to *bb* (start *aa* in column 8).

a*

Lists all screen spaces beginning with character(s) *a* (start *a* in column 8).

The following example lists all screens that have names starting 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

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

.SGO has a conditional screen fetch feature that fetches a screen space only if a condition is true.

Format:

```
n.SGO cccccccc
      [CPSER   {=|EQ|GE|GT|LE|LT|NE} argument]
      [MODE    {=|EQ|GE|GT|LE|LT|NE} argument]
      [PREFIX  {=|EQ|GE|GT|LE|LT|NE} argument]
      [SMFID   {=|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]
```

where

```
n.SGO cccccccc
      [CPSER   {=|EQ|GE|GT|LE|LT|NE} argument]
      [MODE    {=|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]
```

n

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

ccccccc

Specifies the screen space name.

Keywords for conditional setting of variables

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

CPSER

CPU serial number. In the case of an MP, this keyword will compare the supplied CPU serial number with the serial numbers of all CPUs in the complex. If the relational argument is equal (= or EQ), OMEGAMON will fetch the screen space the first time it finds a match. If the relational argument is NE, OMEGAMON will fetch the screen space only after it has checked all of the CPUs in the complex.

MODE

The 3-character code for the OMEGAMON mode of operation. It is displayed on the INFO-line during a session. Refer to [“INFO-line format”](#) on page 12 for a list.

PREFIX

The OMEGAMON product code (for example, OM).

SMFID

The system ID from the SYS= startup parameter or the default SMF ID.

UNIT

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

USER

The user profile suffix from the USER= startup parameter.

&var

Allows you to set any comparison you wish. The keyword and argument can be any variable name set with the .VAR command or any OMEGAMON-defined variable. The .VAR command lists OMEGAMON-defined variables.

Relational operators

The relational operators require blanks on either side except for 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 1- to 8-character value to which OMEGAMON compares the keyword. The keyword and argument can be any variable name set with the .VAR command or any OMEGAMON-defined variable. The .VAR command lists OMEGAMON-defined variables.

For example,

```
.SGO DISK
```

fetches screen space DISK on the next cycle.

```
B.SGO DISK
```

delays the fetch of screen space DISK 11 cycles, and fetches it on the next cycle after $n=0$.

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

You may also use the .FGO command to fetch screen spaces. It functions the same as .SGO except that .FGO bypasses the screen display and the OMEGAMON cycle wait.

.SMF

Displays information about SMF data sets and exits.

Type:

Immediate

The command displays the following information:

```
.SMF SMF System ID is cccc: Job Wait Time Limit tt:tt u +
      nnnn SMF buffers containing nnnnn records have been written
      SMF Dsname volser Status Blocks %full
      dddddddd vvvvvv ssssss nn nn
      SMF exit information:
      Subsystem name: aaaaaa bbbbbb
      eeeeeee ffffffff eeeeeee fffffff eeeeeee fffffff eeeeeee fffffff
```

where

cccc

Is the System Management Facilities (SMF) ID of the system.

tt:tt

Is the Job Wait time (JWT parameter of SMFPRMxx).

u

are units of time. These can be:

SEC

Seconds

MN

Minutes

HR

Hours

ddddddd

Data set Name.

vvvvvv

Volser.

sssssss

Status of the SMF data set. This can be:

Dump reqd

Dump is required

*** active ***

Active data set

alternate

Alternate data set

nn

Decimal value.

aaaaaa

SMF subsystem name.

bbbbbbb

Status of this subsystem. This can be:

*** default ***

Default SMF subsystem.

is inactive

Subsystem is not currently active.

has no exit table

Subsystem does not have an exit table.

exit table has no entries

Subsystem has an exit table, but it is empty.

eeeeeee

Exit name.

ffffff

Status of this exit (active or inactive).

If SMF records are lost, the number of lost records appears at the end of the first line. [Figure 12](#) on page 59 shows a typical SMF display.

```
.SMF SMF System ID is SYSA; Job Wait Time Limit 30:00 MN
+ 1245 SMF buffers containing 44539 records have been written

+ SMF Dsname  volser  Status  Blocks  %full
+ SYS1.MANX  SYSRS1 * active * 4000    90
+ SYS1.MANY  SYSRS2 dump reqd 4000   100
+ SYS1.MANZ  SYSRS3 alternate 4000    0

+ SMF exit information:
+ Subsystem name: SYS      * default *
+ IEFU29 active IEFUTL active
+ IEFUJI active
```

Figure 12. Typical SMF display

If a subsystem has no exits, the following message appears:

```
Exit table has no entries.
```

.SPT

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

Type:

Immediate

The .SPT command can set up to 10 patterns for use with the DEVP device major command and the address space major commands.

Format:

```
.SPT/n ccccccc where
```

n

Specifies the pattern number. It can be a number from 1 to 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 comments out .SPT/C so that it does not re-execute.

..
(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 beginning with OP.

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

The entry

```
DEVP/2
```

invokes pattern 2 for the major command DEVP.

/STK

Saves and stacks the current screen output for later recall.

OMEGAMON currently refreshes the screen every cycle. There are times, however, when you may want to save a screen's output so you can return to it later. You may 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

where ccccc is one of the following arguments for recalling and deleting entries from the stack:

(blank)

Without an argument, /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 prior to the current one. In the .DSE display, the Up argument moves the pointer to the entry over 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 beneath 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 being consumed to stack them. A current entry pointer indicates the most recently referenced screen in the stack.

OMEGAMON displays a message on the INFO-line when it recalls a screen from the stack to indicate that this is not a currently executed screen. If OMEGAMON 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 was on when the screen space was saved and if extended color is on when OMEGAMON recalls the screen. (See the .SCC command in the Profile Menu to activate extended color.)

The following /STK command saves the current screen, M110, onto the stack:

```
/STK_____ M110 TSO OM/DEX V420./C SYS 02/12/09 9:14:21 S
```

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

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

```
/STK 2_____ #03 TSO OM/DEX V420./C SYS 02/12/09 9:14:21 S
```

Figure 13 on page 61 shows a typical result of /STK 2.

				<== 0B0226 DISKS	recalled	11:56:00
DISK	SYSB24	WORKB2	MVSA21	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 13. Result of /STK2

STOP

Stops OMEGAMON.

Type:

Immediate

STOP can be entered either on the INFO-line or in the main body of the screen to stop OMEGAMON.

/TOP

Scrolls to the top of the logical screen.

Type:

INFO-line

TSO

Executes TSO command or CLIST from OMEGAMON.

Type:

Immediate

The TSO command is available only when you run OMEGAMON in TSO mode.

/UP

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

Type:

INFO-line

Format:

/UP nnn

nnnScrolls *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 top of the screen.

PAGE

Scrolls so that the current cursor position is at the top of the screen. This is the default.

TOP

Scrolls to the top of the screen.

The following example scrolls up 20 lines.

/UP 20

If you have assigned 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 interprets the entry as if you had typed the command plus the arguments.

/UP 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]
      [MODE   {|=|EQ|GE|GT|LE|LT|NE} argument]
      [PREFIX {|=|EQ|GE|GT|LE|LT|NE} argument]
      [SMFID  {|=|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]
```

where

C

Requests conditional processing. Allows you to specify a value for a given variable when the criteria is matched.

SET or S

Sets or changes the value of a variable.

&variable

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

value

1- to 64-character alphanumeric string assigned to &variable. Single quotes 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 lists all variables.

DEL

Deletes the specified variable.

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

CPSER

CPU serial number. In the case of a multi-processor environment, the supplied CPU serial number is compared with the serial numbers of all CPUs in the complex. If the relational argument is equal (= or EQ), OMEGAMON will set the variable the first time it finds a match. If the relational argument is not equal (NE), OMEGAMON will set the variable only after it has checked all of the CPUs in the complex.

MODE

The 3-character code for the OMEGAMON mode of operation. It is displayed on the INFO-line during a session. Refer to [“INFO-line format” on page 12](#) for a list.

PREFIX

The OMEGAMON product code (for example, OM).

UNIT

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

USER

The user profile suffix from the USER= startup parameter.

&var

Allows you to set any comparison you want. The keyword and argument can be any variable name set with the .VAR command or any OMEGAMON-defined variable.

OMEGAMON defines the following variables internally for your use:

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

&ZOOM

data found at the cursor location

&ZOOMC

command or exception name field

&ZOOMS

originating screen space

- Exception variables:

&ZXccccT

threshold value

&ZXccccV

last trip value

&ZXccccW

worst trip value

The variable cccc is the exception name.

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

The relational operators require blanks on either side except for the equal sign (=).

=

Keyword equals argument. The equal sign can be entered without blanks on either side of it.

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 1 to 8-character value to which OMEGAMON compares the keyword. OMEGAMON variables are permitted. The keyword and argument can be any variable name set with the .VAR command.

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

For example:

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

set values for NXTSCRN and DEVICE. You can now use those variables in a screen space.

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

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

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

In this example, the variable *SYSTEM* will be set to A if the variable *SWITCH* has previously been set to YES; *SYSTEM* will be set to B if *SWITCH* has previously been set to NO; and *SYSTEM* will be set to C if *SWITCH* has previously been 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* will be set 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 logged onto OMEGAMON in VTAM mode.

Type:

Immediate

This command allows a user running in a multi-session environment 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 show up.

Here is an example of the display.

```
.VTM  Userid  Terminal  Mode      Session Start      Last Update
+     TS0X07  L65F      VTM       02/12/09 11:45:32     02/12/09 13:14:55
+     TS0X06K BBLVM06   VTT       02/12/09 13:57:32     IN INITIALIZATION
+     AF0PER  BBLVM04   VTT       02/12/09 11:43:22     02/12/09 13:57:12
+     TS0X21  L616A09   VTS       02/12/09 10:23:31     02/12/09 11:24:31
+     TS0X36A L674      VTM       02/12/09 13:40:33     02/12/09 13:57:51
```

The Mode field indicates the type of session. The possibilities are:

VTM

A VTAM mode session.

VTS

An ISPF mode session running under VTAM.

VTT

A TSO mode session running under VTAM.

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

.WAT

Waits *nn* seconds before executing all following commands.

Type:

Immediate

Format:

.WATnn

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

The following example submits a job, then waits 10 seconds to check the status of the job.

```
TSO SUBMIT 'USER1.JCL.CNTL(JOBA) '
.WAT10
TSO STATUS
```

/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.

This feature requires DEXAN.

For information about using the XLF feature, see [Appendix A, "Automating and logging features," on page 309](#).

.XLFOUT is the equivalent immediate command.

.XLFOUT

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

Type:

Immediate

.XLFOUT comments itself out after it executes.

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 .XLG, a series of keywords appears for setting printer characteristics. If you are changing more parameters than will fit on one line, use the OUTF command instead. The keywords for .XLG (and .REP) duplicate the minors of the OUTF major.

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 will also appear if their default values have been 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:

OMEGAMON generates separator pages and page headers with the appropriate job name printed in block letters on the pages. This is the default.

NONE

OMEGAMON does not generate page headers or separator pages.

ccccccc

OMEGAMON generates separator pages and page headers with ccccccc printed in block letters on the pages. ccccccc is up to 8 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 ddnames.

If you change any parameter other than FOLD=, OMEGAMON 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.

Parameters	Default values	
	SYSOUT	DDNAME
SYSOUT=	A	(inactive)
HOLD=	NO	(inactive)
COPIES=	1	(inactive)
FORMS=	*NONE*	(inactive)

Parameters	Default values	
	SYSOUT	DDNAME
DEST=	*NONE*	(inactive)
DDNAME=	(inactive)	OMXLFLOG
FOLD=	YES	YES
ID1=	jobname	jobname
ID2=	(blank)	(blank)
ID3=	(blank)	(blank)
ID4=	(blank)	(blank)

.ZAP

Displays maintenance ZAPs applied.

Type:

Immediate

/ZOOM

Invokes navigational zoom feature 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 over the current cursor position for a variable 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 enables quick investigation of exception conditions.

Note the following points regarding this feature.

- IBM ships OMEGAMON with the /ZOOM INFO-line command assigned to PF11 so you can access the zooming screen spaces with a single key. We refer to PF11 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 zooms to a recommendation screen that gives you suggestions on actions you may 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 a navigation option.

```

----- ZOPT   TSO   OM/DEX  V420./C SYS  02/12/09 15:46:07
> HeIp PF1   Back PF3      Up PF7      Down PF8      Zoom PF11
=====

```

Note: If you want to be able 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 found at the cursor location.

&ZOOMC

Command or exception name field (columns 2 to 5).

&ZOOMS

Originating screen space.

For information on how to create screen spaces, see [Chapter 4, "Screen spaces,"](#) on page 71.

OMEGAMON provides some sample zooming screen spaces to help you become familiar with both the setup of a zooming screen space and the types of commands for which /ZOOM is most appropriate. To list the names of those screen spaces, use the SCRN command and look for screen spaces named in the format @ZOMcccc where cccc is a command name. Then try zooming on those command names.

Here is an example of using the zooming feature. You can follow this example either in command mode by entering the DSKB command or in menu mode by choosing the I/O DASD option.

1. Execute the major command DSKB. DSKB lists the unit addresses of disks that are currently performing I/O.

```
----- BUSYDISK TSO OM/DEX V420./C SYS 02/12/09 12:06:26
<DSKB SYSA22 MVSA21 WORKA1 OMON26
```

2. To examine one of the disks more closely, place your cursor under the volser you wish to query (in this case MVSA21) and press PF11. OMEGAMON looks for a screen space starting with @ZOM and ending with the major command name that precedes the cursor. OMEGAMON finds the screen space @ZOMDSKB.

```
----- @ZOMDSKB TSO OM/DEX V420./C SYS 02/12/09 12:06:26
<DSKB SYSA22 MVSA21 WORKA1 OMON26

DEV &ZOOM
dadr
dio .R
dioq
dusr
dplt02
```

/ZOOM replaces the variable &ZOOM with the volser over the cursor. The result might look like this:

```
----- @ZOMDSKB TSO OM/DEX V420./C SYS 02/12/09 12:06:26
<DSKB SYSA22 MVSA21 WORKA1 OMON26

DEV 150 volser=MVSA21 Online Alloc
dadr 150
dio .R 34.4
dioq
dusr CATALOG
dplt02 ----- OMEGAMON Peek at Unit=150 volser=MVSA21 -----
+ |DBsy DDD DDD D DD D D DD D DD | Samples: 50
+ |IOQ | Interval: 7
+ |I/O# >54>58>60>62>65 >67>69>73>77>79 | I/O's: 26
+ |CPU .....|
+ |User >CATALOG |
+ |Cyl >0292>0006 >0292 >0006 |
+ |Nrdy |
+ |Resv |
```

3. You could now place your cursor on a different volser and press PF11 again to receive the same detail on another device. A zooming screen space can display more detailed analyses of any device, volume serial number, or address space.

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

b

blank

+ plus sign
() left and right parenthesis
| vertical bar
' single quote
<> greater than and less than sign
= equal sign
* asterisk

Chapter 4. Screen spaces

Through its powerful screen space feature, OMEGAMON stores, recalls, and invokes a set of commands. IBM ships OMEGAMON with a number of predefined screen spaces, which are designed to give you detailed information by category.

This section contains the following topics related to using screen spaces:

- [“Invoking screen spaces” on page 71](#)
- [“Creating or modifying screen spaces” on page 71](#)
- [“Loading screen spaces” on page 73](#)
- [“Listing screen spaces” on page 73](#)
- [“Renaming a screen space” on page 73](#)
- [“Deleting a screen space” on page 74](#)
- [“Invoking screen spaces automatically” on page 74](#)
- [“Using screen spaces with the /ZOOM command” on page 75](#)

Invoking screen spaces

You can invoke a IBM-defined screen space by making a menu selection, or by pressing certain PF keys if you are working in command mode. You can also create custom screen spaces and recall them by typing the screen space name on the INFO-line, or by assigning the screen space name to a PF key.

When you invoke a screen space, its commands replace the commands currently on the screen. The commands then execute the same as they do when you enter them individually. You must set up screen spaces in advance in order to use the OMEGAMON Automatic Screen Facility, the Timed Screen Facility, and the zooming feature.

Creating or modifying screen spaces

You can create a screen space at any time on any screen by simply typing in a group of commands, assigning a name to them, and saving them.

Note the following points about creating screen spaces:

- When you create a screen space, you can include separator lines to make the screen more readable. OMEGAMON automatically draws a separator line across the screen for you with the `c====aa` immediate command. The variable `c` allows you to specify the color if you have extended color on your terminal, and the variable `aa` allows you to specify the characters you want to use for the line.
- Use comment lines, wherever appropriate, to explain the commands. Begin all lines of comment text with a greater-than sign (`>`) in column 1.
- If you include in the screen space any commands that comment themselves out after execution, make sure that the comment character (`>`) does not appear in column 1 when you save the screen. To prevent this, use the definition mode (described later in this section) while you design the screen.
- You *cannot* include INFO-line commands as part of a screen space. Instead, use the equivalent immediate command.
- OMEGAMON does not save blank lines at the bottom of a screen space.
- The default cursor position is the first position of the INFO-line. However, if you want to save the cursor in a particular position on the screen, type `/REP` or `/SAVE` on the INFO-line, move the cursor to the desired position, and then press Enter to execute the command. The cursor will display in that position whenever OMEGAMON invokes the screen.
- You can use a standard text editor to create or alter screen spaces. If you look at or edit a screen space using a text editor, be aware of the following:

- If you have placed the cursor in any position besides the default position, the INFO-line displays CURS=(n,m) where the variable n is the cursor row position and m is the column position.
- You can add or change the cursor position in a saved screen using a text editor by adding or changing CURS=(n,m).

The following example shows a screen space called SAMPLE. Notice the command explanations in the divider lines and the additional comments in the note at the bottom of the screen.

```

----- SAMPLE TSO LOG OM/DEX V420./C SYS 02/12/09 08:31:41
>***** SAMPLE - System Paging Rates *****
  SYS
>===== Paging: Total system (SPAL = SPS + SPSV + SPWS)=====
>
$SPAL.R
>
>===== Paging: System area (private and common) =====
>
$SPS .R
>
>===== Paging: Due to swapping =====
>
$SPWS.R
>
>===== Paging: Total common area =====
>
$SPC .R
>
>===== Paging: Total common area page-ins =====
$SPCI.R
>
>NOTES: To change the scale of the plot, use the GSCL immediate command

```

Figure 14. A sample screen space

OMEGAMON accepts variable symbols as arguments for commands. The .VAR immediate command allows you to set variable symbols for use in designing screen spaces. You can define the variables at OMEGAMON startup or at any other time during your session. For more information on using variables, see the .VAR command.

To create or modify a screen space, follow this procedure:

1. If you are creating a single screen space, enter the .DEFON immediate command or the /DEF ON INFO-line command to place OMEGAMON in definition mode until you are done.

Definition mode inhibits updating and prevents OMEGAMON from executing the commands you have entered.

Because saving the screen space automatically turns off definition mode, use /DEF HOLD (or .DEFHO) if you are creating several screen spaces at once. This holds definition mode until you issue a /DEF OFF or .DEFOFF command.

2. Enter the desired commands on the screen.
3. Use the /SAVE INFO-line command to save the screen. If you want to replace an existing screen space with a new screen space of the same name, use the /REP command.
4. If you have turned on definition mode with the HOLD option, use .DEFOFF or /DEF OFF to restore automatic updating.

Loading screen spaces

The LSCR immediate command loads screen spaces from disk storage to main storage. By placing screen spaces in storage with LSCR, you make them more available and more easily fetched. If a disk is not available, you can continue to invoke the screen spaces that you loaded into main storage with LSCR.

The format is:

```
LSCR cccccccc cccccccc || cccccccc
```

The variables are screen space names. Specify screen space names starting in column 8. You can load as many screens as can fit on the input line, up to the last eight columns. In the following example, OMEGAMON attempts to load screen spaces ZZ1, ZZ2, and ZZ3 from disk to main storage.

```
LSCR ZZ1 ZZ2 ZZ3
```

Listing screen spaces

The SCRNB immediate command, used without an argument, lists all screen spaces in main storage and in the disk data sets pointed to by the RKOMPCSV and RKOMPROC ddnames. You can limit the list by using arguments to request a list of screens in main storage only, disk only, or both. You can also specify alphabetic limits for a range of member names, and wild cards are allowed.

Figure 15 on page 73 shows a typical default screen space listing. OMEGAMON sorts the list by data set, with in-storage screens first, followed by RKOMPCSV, then each data set in the RKOMPROC concatenation.

```
>SCRNB $ thru 99999999
> In-storage screen facility 20 members 4232 bytes used
> $ #01 #02 #03 DEBUG DEBUG01 DEBUG02 DEBUG03
...
> RKOMPCSV rhilev.RKOMPCSV
> #01 SAMPLE VSAM
> RKOMPROC (DUP.) rhilev.RKOMPCSV
> RKOMPROC +01 thilev.RKOMPROC
> #01 #07 #08 #09 #10 #13 #14 #15
> #16 #17 #18 #19 #20 #21 #22 #23
> #24 BITMAP CPAM CSA DD DDACVOPS DDBASIC DDDEX
> DDEXCEPT DDIO DDNAMES DDNEWS DDOMEGA DDOPER DDPFK DDSYS
> DDSYSE DDSYSI DDTASK DEBUG DEBUG01 DEBUG02 DEBUG03 DEBUG04
> DEXALL DEXAL2 DEXAL3 DEXAL4 DEXAL5 DEXAN DEXDFLT DEXDF2
> DSA ENQUEUEUES EOD HELP HELP2 INTERVAL OMINIT00 PAM
> SCREEN SCREENS SIT SLOT SLOTS STOP STOPC STOPI
> SUBPOOLS TABLES TASK TASKS VIRTUAL VSAM
```

Figure 15. Screen space list

Renaming a screen space

Use the RENM immediate command to rename a screen space. The RENM command accepts an argument specifying where the screen space is to be found.

If you want to rename the SAMPLE screen space to EXAMPLE in both main storage and RKOMPCSV, enter:

```
RENMB SAMPLE EXAMPLE
```

The following message appears:

```
> >> Member "SAMPLE " Renamed to "EXAMPLE " Both In-Storage and in RKOMPCSV <<
```

The screen space SAMPLE no longer exists.

Deleting a screen space

Use the DELT immediate command to delete screen spaces. OMEGAMON deletes screen spaces only from main storage and the data set pointed to by the RKOMPCSV ddname, not from the data set pointed to by the RKOMPROC ddname, the read-only data set. The DELT command accepts an argument specifying where the screen space is to be found.

If you wish to delete the screen space SAMPLE from both main storage and RKOMPCSV, enter:

```
DELTB SAMPLE
```

The following message appears:

```
> >> Member "SAMPLE " Deleted Both In-Storage and from RKOMPCSV <<
```

Invoking screen spaces automatically

OMEGAMON offers the capability of chaining together a group of screen spaces, so that an entire series of screens can execute without operator intervention. An .SGO or .FGO immediate command, entered in a predefined screen space, fetches the next screen automatically. The .FGO (fast go) and .SGO (screen go) commands perform identical functions, but .SGO displays the screens as they execute, whereas .FGO bypasses the OMEGAMON cycle and the terminal display.

The screen space fetch feature is particularly useful in the OMEGAMON Automatic Screen Facility (ASF) or Timed Screen Facility (TSF). ASF allows your system to react automatically to unexpected problem conditions in the system. It can invoke pre-defined screen spaces containing commands to investigate the condition in detail and log the output. TSF enables you to schedule monitoring tasks at regular intervals. See [Appendix A, "Automating and logging features,"](#) on page 309 for an explanation of how to set up ASF and TSF.

Conditional screen space fetching

The .FGO and .SGO commands give you the flexibility of conditional screen space fetching.

You can enter an argument, which may include relational operators, to fetch a screen space only if certain conditions are true. You can compare the CPU serial number, the mode of operation, the operating system level, the profile in use, the SMF ID, the OMEGAMON console address, or any variable you have set with the .VAR command.

Delayed screen space fetching

The .SGO command lets you delay the fetching of a screen space for a specified number of OMEGAMON cycles. Delaying the fetching allows you to accommodate commands that take more than one cycle to initialize (gather data). It also permits you to log several successive executions of a given screen space before moving on to the next screen space.

A label in front of .SGO specifies the delay.

```
n.SGO screenname
```

The variable *n* can be a number representing a delay of 1 to 9 cycles or a letter from A to Z representing a delay of 10 to 35 cycles.

Looping screen spaces

An improper sequence of .FGO screens could cause a looping condition. To protect against this possibility, OMEGAMON limits the number of consecutive fetches allowed (64 by default). After reaching the limit, .FGO acts like .SGO so that executing screen spaces display on each cycle. Therefore, once .FGO screen spaces cause a loop, you need to correct the condition and re-enable .FGO.

To re-enable .FGO, use the FGLOOP keyword of the .SET command. The FGLOOP is set to OFF until the limit is reached. Then OMEGAMON turns it ON, indicating the probability of a loop.

To reset .FGO, issue the .SET command and replace the ON argument with OFF.

Testing .FGO routines

Because FGLOOP=ON causes .FGO *not* to bypass the OMEGAMON cycle and the terminal display, you may want to turn it on yourself to test your screen space fetch routines.

Manipulating the log within screen spaces

To enable you to log screen spaces selectively, the .LOG immediate command has PUSH and POP arguments. You can include these arguments in screen spaces to change the on or off status of the log when you branch to other screen spaces, and then return the log to its original state after the screen routines are completed.

For example, if you had a series of four screen spaces chained together with .FGO and you wanted to log only the last two, you could use .LOGPUSH and .LOGPOP in the following manner:

1. At the end of screen space 2, enter .LOGPUSH to have OMEGAMON record the previous status of the log. Then, enter OPTN LOG=ON to turn on the log at the beginning of screen space 3.
2. Enter .LOGPOP at the end of screen space 4. Thus, the log will be restored to its original status when your branching routine is complete.

The .LOGPUSH and .LOGPOP functions are performed automatically when you log screen spaces using the Exception Logging Facility (XLF), the Automatic Screen Facility (ASF), or the Timed Screen Facility (TSF).

Using screen spaces with the /ZOOM command

There are a number of guidelines you should follow when you set up screen spaces for use with the /ZOOM command.

- The screen space name must start with a four-character prefix that begins with an alpha or national character (@ZOM is the default) and is followed by the command or exception name (@ZOMcccc).
- When creating your own zooming screen spaces, you can name them with your own prefix (aaaa) instead of @ZOM.

To invoke a zooming screen space that has your aaaa prefix, do one of the following:

- Type /ZOOM aaaa on the INFO-line, move the cursor to the data, and press Enter.
- Type aaaa on the INFO-line, move the cursor to the data, and press the Zoom key.
- Redefine the Zoom key, PF11, or assign another PF key to /ZOOM aaaa, move the cursor to the data, and press the Zoom key.

Chapter 5. The User Profile Facility

OMEGAMON profiles control the characteristics of an active OMEGAMON session. There are three types of profiles: IBM-supplied, installation, and individual user. Installation profiles are intended to be created by the installer (or other authorized personnel) and specified as the default for the site or for a system. There can be as many individual profiles as you want. Users can create them, save them by assigning them a unique 2-character identifier, and specify their desired profile ID at logon.

Both installation profiles and individual user profiles are created by customizing the settings of 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 setting of user profile options at any time during a session, and they take effect immediately.

The following table summarizes the profile definition commands.

Command Function	Command	Description
Controlling session and display options	OPTN	<p>Activates and deactivates the:</p> <ul style="list-style-type: none">• Automatic Screen Facility (ASF)• Timed Screen Facility (TSF)• Exception Logging Facility (XLF)• Terminal bell• Log <p>Controls these display characteristics:</p> <ul style="list-style-type: none">• The date (USA or European)• Minor commands (upper or lower case)• All command output (upper or mixed case)• Scroll amount (page or cursor)• The first screen when OMEGAMON is started• The interval for the terminal bell <p>Note: The setting for the ZEROS keyword is not saved in a profile.</p>

Table 4. Profile definition (continued)

Command Function	Command	Description
Setting print output options	OUTP	The minor command settings are saved in a profile: <ul style="list-style-type: none"> • COPY specifies number of copies to print. • DDNM specifies ddname to override standard ddname. • DEST specifies the report destination. • DSTU specifies the userid for a report. • FOLD specifies whether lowercase is folded to uppercase. • FORM specifies the name of the form on which to print. • HOLD specifies that the output be placed in the hold queue and retrieved from TSO. • ID1 requests separator pages and page headers that identify output from different OMEGAMON sessions. • 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. • LNCT sets number of lines per page on report. • SOUT defines the SYSOUT class of the report.
Setting color options	.SCC	Allows you to set color, highlighting, and extended attribute options for each field on the OMEGAMON display.
Setting operational parameters	.SET	Sets parameters for: <ul style="list-style-type: none"> • The screen space fetch feature (.FGO) • The interval for OMEGAMON cycles • The number of entries in the device name table • Guarding against loops 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.
Customizing tables	XQN	Lets you name the enqueues that you want to exclude from enqueue conflict analysis.
Setting installation profile options	IOPT	Determines whether OMEGAMON storage is page-fixed and whether OMEGAMON issues a DASD RESERVE when users save a screen space member.
Defining Timed Screen Facility entries	.TSF	(See Appendix A, "Automating and logging features," on page 309.)
Setting exception analysis options		(See table in Chapter 6, "Exception analysis," on page 93.)
Setting OMEGAMON product options	POPT	Controls how certain commands or exceptions operate. Commands: MCPU, SCPU, XDSK, NCLV minors. Exceptions: DYPR, VTOC

Multiline input

Some profile-definition commands use a multiline input facility (for example, OPTN, .SCC, and .SET). When you enter these commands, OMEGAMON displays the current settings for their 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 re-displays the line on the next cycle.
- To change a setting, type over the displayed value and press Enter.

Profile maintenance commands

The profile maintenance commands are IPRF (for the installation profile) and PPRF (for user profiles).

IPRF

Saves or deletes the installation-defined default profile.

Type:

Immediate

Format:

IPRF SAVE|DELETE

SAVE

Saves the installation profile in the data set defined by the RKOMPFSV DD statement.

DELETE

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

OMEGAMON automatically assigns the 2-character code /I to the installation profile. /I is used on the USER= startup parameter to load this profile, and it appears on the INFO-line during the session.

The installation-defined profile for OMEGAMON for z/OS is stored as member name OMINSTAL.

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 RKOMPFSV DD statement.

DELETE

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

COMMENT

Displays a description of the current profile and allows you to change it dynamically.

LIST

Lists all members of the user profile data set.

cc

Specifies the 2-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

Parameters of the OPTN command 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   = OMINITZZ     LOG           = OFF
:   MINORCASE     = LOWER        SCREENCASE    = MIX
:   SCROLL        = PAGE         TSF           = OFF
:   XLF           = OFF          ZEROS         = OFF
```

ASF

Turns the Automatic Screen Facility (ASF) ON or OFF.

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 display.

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.

XLF

Turns the Exception Logging Facility (XLF) ON or OFF.

ZEROS

Sets the way in which zeros display on your terminal. When this parameter is ON, OMEGAMON displays the numeral 0 in fields that have a value of zero. When it is OFF, OMEGAMON 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 ccccc

The variable ccccc is either XLFLOG or REPORT.

The OUTP major command displays column headings for pending and current values associated with all of its minor commands:

```

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

```

If 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 will become the active value when you reallocate the log with /LOG OUT, .LOGOUT, /XLFLOG, or .XLFLOG.

The log will be routed to one of the following data set types:

- SYSOUT, the initial data set type for OUTP. This file is designated FREE=CLOSE. This 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. If you specify attributes that do not conform to these standards, OMEGAMON will dynamically reset them.

The IDSP parameter may be set to MOD, SHR, or OLD.



CAUTION: The OMREPORT DD statement is optional and is *not recommended*. If DISP=SHR or DISP=OLD is used, any action to close and reopen the file will reinitialize it, thus deleting existing information.

Only one session at a time per address space can use a specific ddname.

Note: Some prior releases of OMEGAMON used ddname OMREPORT if it was present. This release also looks for ddname OMREPORT if present, but will access 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 currently inactive. [Table 5 on page 81](#) shows the minors and the initial settings for the SYSOUT or ddname data sets.

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

<i>Table 5. OUTP minor commands and initial values (continued)</i>		
Parameters	Initial values	
	SYSOUT	DDNAME
ID1	jobname	jobname
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 ignores the COPY command. If the value you type is not valid, OMEGAMON redisplay it where you typed it and does not transfer it to the pending column.

DDNM

Directs output to a particular ddname.

The DDNM minor removes control from SYSOUT.

Type:

Minor of OUTP

Format:

DDNM ccccccc

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

DEST

Specifies the destination that is to receive the output.

The DEST minor is under SYSOUT control.

Type:

Minor of OUTP

Format:

DEST ccccccc

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 installation determines valid destinations. OMEGAMON checks your installation's table for valid destinations only when you reset the log (with .LOGOUT or .XLFOU). During initialization, OMEGAMON only checks 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 ccccccc

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

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

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

If the value you type is not valid, OMEGAMON redisplay 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 be 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 redisplay 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 sessions.

ID1 is active under ddname or SYSOUT.

Type:

Minor of OUTP

Format:

ID1 ccccccc

OMEGAMON generates separator pages and page headers with the appropriate job name printed in block letters on the pages. This is the default.

NONE

OMEGAMON does not generate separator pages and page headers.

ccccccc

OMEGAMON generates separator pages and page headers with ccccccc printed in block letters on the pages. ccccccc is up to 8 user-defined characters.

ID2

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

ID2 is active under ddname or SYSOUT.

Type:

Minor of OUTP

Format:

ID2 ccc|ccc

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

ID3

Defines up to 16 characters in the center of 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 appear centered on the separator page beneath the block letters.

ID4

Defines up to 16 characters on 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 appear on the right of the separator page justified below the block letters.

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 currently used for your XLFLOG or REPORT log and directs it to SYSOUT.

Type:
Minor of OUTP

Format:
SOUT c

The variable *c* is the SYSOUT class.

Setting color options

The .SCC command controls color or highlighting.

.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 screen is sent to a user's terminal. Since display characteristics depend on the type of terminal you are supporting, .SCC contains keywords that, in combination, will accommodate any of the various 3270-type devices. Here is the screen display format:

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

Device support

This discussion divides color/highlighting capabilities into the following device categories:

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, which use high or low intensity to distinguish fields in a display.
- Color non-EDS, which distinguish between high and low intensity by using different colors in a display. The colors displayed depend on whether the device is a 2-base color or 4-base color device, and whether a field is protected or unprotected. For example, most 4-base color devices display high intensity fields in red (unprotected) and white (protected), low intensity fields in green (unprotected) and blue (protected).

Monochrome EDS

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

Color EDS

Color devices that support the extended data stream. A field may be displayed in any of the 7 extended colors in combination with the extended highlighting attributes.

Profile definition mode

Before customizing your color/highlighting definitions, determine which mode of .SCC is appropriate. The mode is controlled by the `ProfileDefinitionMode` keyword:

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

OFF

Changes to .SCC affect only the current OMEGAMON session.

ON

Intended for use when creating or changing a user profile. After defining the settings, you must issue a profile save command in order for these definitions to be saved and reused. (They may also take effect during the current session if you subsequently change the value of the `Display=` keyword.) Setting this keyword to ON enables you to configure options for different types of terminals on the same screen.

Keywords and values

For any keyword or value, you need only type as many letters as it takes to make an entry unique. You may type any keyword and value on the command line after .SCC and bypass the multiline display. 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.

Display intensity or color

The `Display` keyword has 4 possible settings.

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

BASIC

Allows you to 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 will treat that device as a four-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, allows you to specify the color of each field.

Note: In cases where OMEGAMON 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 may get a `PROGnnn` or a screen erasure error. You may press the `ATTN` or `PA1` key to resume the session, but be aware that this action also clears the current security authorization and the current screen space.

Display fields

Following is the display format of `.SCC` with `ProfileDefinitionMode=ON`. It shows the display field keywords and valid values.

```
.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
```

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 the figure, the Default value is abbreviated as DEF.

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

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 (Red, Blue, and so on) or highlighting (HI or LO) when customizing exception messages with the XACB command.

The following table shows the possible variations of settings according to the type of terminal you are supporting.

<i>Table 6. Color and highlighting settings in .SCC</i>		
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

The .SET immediate command displays data about the OMEGAMON environment in table form. You can modify or change data on any of the output lines.

.SET

Sets operation control parameters.

Type:

Immediate

Format:

```

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

```

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

FGOLIMIT

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

FGOLOOP

Turns .FGO screen loop detection ON or OFF. If OMEGAMON detects an .FGO loop, it sets this keyword to ON and then treats subsequent .FGO commands as .SGO commands. (.FGO executes a screen space without displaying it; .SGO displays each screen space it executes.) You can also set .FGOLOOP=ON yourself if you want to test screen spaces you have linked 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 time between automatic updates (called an OMEGAMON cycle). The following restrictions apply:

- This value will be effective only in dedicated mode or VTAM mode. You can, however, be operating in any mode when you define a new value for the purpose of saving it in a user profile.
- The maximum interval is 99.00.
- VTAM mode does not allow an interval shorter than 5 seconds.
- Dedicated mode does not allow an interval shorter than .5 seconds.

IODELAY

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

LOOPCOUNT

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

LOOPTIME

The threshold (in seconds) for OMEGAMON built-in loop detection. The default is 10 seconds, the maximum value is 300.

PAGELIMIT

The size (in pages) of the REPORT file used to log OMEGAMON screens. The maximum is 99999. This number dynamically decreases as the log is printing to reflect the number of pages left before the limit is reached. Consequently, be sure to check this parameter (and reset it, if necessary) before saving a profile.

PEEKSIZE

The size (in bytes) of the PEEK buffer. The maximum is 204800.

STATUSMODE

The status mode (ON or OFF) for the Workload Profile Facility.

Exclude Enqueue Names command

The XQN command lets you define, by name, the enqueues that you want excluded from enqueue conflict analysis. Excluding some enqueue names helps to suppress extraneous lines in the XXQCB command display. The XQN command uses the multiline input facility to make it easy for you to change displayed values.

Type:

Immediate

The following considerations apply to the multiline input facility:

- You can modify any display line that begins with a colon in column 1.
- Blanking out fields has no effect; OMEGAMON redisplay the line on the next cycle.
- To change a setting, type over the displayed value and press Enter.
- OMEGAMON marks modified entries with one of the following words:

```
ADDED
UPDATED
DELETED
```

OMEGAMON treats commas, blanks, and parentheses in command syntax as delimiters.

If you issue the XQN command without any operands or keywords, OMEGAMON displays either the first 100 entries or the entire enqueue name table. [Figure 16 on page 90](#) shows a typical enqueue name table.

```
-----ZWELCOME TSO OM/DEX  V420./C SYS 02/12/09 16:46:01
XQN
+ #      ENQ NAME
: 1      SPFEDIT
: 2      SPFDSN
: 3      SYSDSN
+ There are 3 entries defined in the ENQ name table.
```

Figure 16. Typical enqueue name table

#

Entry number in the enqueue name table.

ENQ NAME

Major name of the enqueue.

If you issue the XQN command with the DIS operand, OMEGAMON displays the enqueue name table in the preceding format (if you do not specify any operands, OMEGAMON assumes the DIS operand). The syntax is:

```
XQN DIS
```

If you issue the XQN command with the ADD operand, you can add one or more enqueue names to the table. The syntax is:

```
XQN ADD name1,name2,...namen
```

where

name

Specifies the major name of an enqueue.

[Figure 17 on page 91](#) shows typical output from the XQN ADD command.


```

----- ZWELCOME TSO OM/DEX V420./C SYS 02/12/09 16:46:01
>XQN ADD SYSZVTOC SYSZVSAM
+ 4   SYSZVTOC * added *
+ 5   SYSZVSAM * added *
+Entries numbers may have been shifted due to the ADD/DEL
+ #   ENQ NAME
+ 1   SPFEDIT
+ 2   SPFDSN
+ 3   SYSDSN
+ 4   SYSZVTOC
+ 5   SYSZVSAM
+ There are 5 entries defined in the ENQ name table.

```

Figure 17. Typical output from the XQN ADD command

After you add entries to the table, OMEGAMON renumbers all entries consecutively and displays an informational message informing you that the entry numbers may have shifted.

If you issue the XQN command with the DEL operand, you can delete one or more enqueue names from the table. The syntax is:

```

XQN DEL [name1,name2,&concat;namen]
        [ENTRY(n1,n2,&concat;nn|n1 : n2)]
        [ALL]

```

where

name

Specifies the major name of an enqueue.

ENTRY

Specifies a list or range of enqueue table entries by number.

ALL

Specifies all entries in the table.

Figure 18 on page 91 shows typical output from the XQN DEL command.

```

----- ZWELCOME TSO OM/DEX V420./C SYS 02/12/09 16:46:01
>XQN DEL E(2,5)
+ 2   SPFDSN * deleted *
+ 5   SYSZVSAM * deleted *
+Entries numbers may have been shifted due to the ADD/DEL
+ #   ENQ NAME
+ 1   SPFEDIT
+ 2   SYSDSN
+ 3   SYSZVTOC
+ There are 3 entries defined in the ENQ name table.

```

Figure 18. Typical XQN DEL command output

After you delete entries from the table, OMEGAMON renumbers all entries consecutively and displays an informational message informing you that the entry numbers may have shifted.

Chapter 6. Exception analysis

Exception analysis is one of the most powerful features of OMEGAMON. An exception is an unusual condition or situation which may affect availability or performance. Exception analysis produces a warning message when a parameter goes above or below a specified threshold. To help you decide upon the appropriate response, OMEGAMON provides suggested screens for each exception.

OMEGAMON is shipped with default thresholds for each exception. However, since each z/OS environment is different, you will want to adjust various exception thresholds to suit the needs of your installation.

Two other OMEGAMON features, the Exception Logging Facility (XLF) and the Automatic Screen Facility (ASF), provide automatic handling of exceptions. See [Appendix A, “Automating and logging features,”](#) on page 309 for details of how to use XLF and ASF.

The information in this section is divided into two parts. The first part describes the immediate commands that control exception analysis. The second part describes individual exceptions and the messages displayed when they exceed their thresholds. See also [“CSA Analyzer-related exceptions”](#) on page 182.

Controlling exception analysis

With the exception analysis immediate commands, you can control exception analysis and display current status.

The following table summarizes the functions of these commands.

Command Function	Command	Description
Starting exception analysis	EXSY XAS XGRP	Invokes exception analysis for all exceptions. Invokes exception analysis for address space exceptions. Invokes exception analysis for the specified exception group.
Summarizing exception activity	XSUM XTRP	Displays a summary of the last and worst trip values for all exceptions. Displays a summary of the last and worst trip values for tripped exceptions.
Defining exception characteristics (Definitions for these commands can be saved in a profile. See Chapter 5, “The User Profile Facility,” on page 77.)	GDFN	Displays existing groups (both IBM-defined and user-defined).
	LEXC XACB	Allows you to define new groups. Allows you to specify the order for OMEGAMON to sample exceptions and display messages. For each individual exception, XACB allows you to: <ul style="list-style-type: none">• 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 sampling.• Define parameters for XLF and ASF.
	XTXT	Allows you to define the message to display when no exceptions have tripped.
Controlling groups	XGSW	Allows you to control the state (ON, OFF) of exceptions by group.

Table 7. Exception Analysis control commands (continued)

Command Function	Command	Description
Defining groups or mask entries for monitoring (Definitions for these commands can be saved in a profile. See Chapter 5, "The User Profile Facility," on page 77.)	ASG	Lets you define groups of address spaces which can be treated as a unit for setting exception thresholds.
	CHNM	Lets you define a channel path mask for use with the missing channel path exception (XCHN).
	CPUM	Lets you define the CPU mask for XCPU, the missing CPU exception.
	MTA	Lets you define the tasks you want to monitor for missing task analysis (XMTA exception).
Setting special thresholds (These threshold definitions can be saved in a profile.)	TPWT	Defines the tape mount wait time threshold for the WAIT exception.

Starting exception analysis

EXSY

Executes OMEGAMON exception analysis.

Type:

Immediate

This command invokes over 50 exceptions.

There are five groups of exceptions:

AS

Auxiliary Storage Manager (ASM)

HD

Hardware

OP

Operations

RS

Real Storage Manager (RSM)

SR

System Resource Manager (SRM)

To execute only the exceptions for a single group, use the XGRPcc command.

EXSY invokes all system-wide exceptions as well as address space exceptions. To invoke only the address space exceptions, issue the XAS command.

A G in column 1 displays the Address Space Threshold (AST) group number (if any) for each address space. The ASG immediate command defines address space threshold groups. These groups allow you to vary the address space exception thresholds for different groups of address spaces.

The output of EXSY only shows the exception message. You may want to change an exception threshold. To see which analysis generated the exception, enter L in column 1 of EXSY. You can enter the 4-character exception name that appears on the LIST=parameter of the XACB command to change the exception threshold.

If you enter EXSY in the form EXSYnn or EXSnnn, the first nn or nnn lines of output are suppressed. If no exceptions have tripped on the current cycle, OMEGAMON displays a message to that effect. You can customize the message with the XTXT profile command.

XAS

Executes address space exception analysis only.

Type:

Immediate

Executes only the exception processors that analyze at the address space level, as opposed to those that analyze at the system level.

XGRP

Invokes exception analysis for exception group *cc*.

The XGRP command invokes only those exceptions belonging to the specified group.

Type:

Immediate

Format:

XGRP*cc*

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

AS

Auxiliary Storage Manager exceptions

HD

hardware exceptions

OP

operations exceptions

RS

Real Storage Manager exceptions

SR

System Resource Manager exceptions

The exception group type codes can be supplemented or replaced by new user defined groups.

Defining exception characteristics

LEXC

Sets order of exceptions for exception analysis sampling.

Type:

Immediate

The LEXC command displays the order in which OMEGAMON executes exceptions. To change the sequence of exception messages displayed by the EXSY command, you can dynamically reorder the execution sequence. Figure 19 on page 95 shows a partial LEXC display. To change the order of the exceptions, type over an exception name or its number.

```

LEXC
:   XECS=1      DRDY=2      VTOC=3      XMTA=4      XGTF=5
:   DYPR=6      XDPR=7      XCSA=8      XACP=9      XOLT=10
:   XMEM=11     XSQA=12     XGRS=13     XDDR=14     PAOD=15
:   WAIT=16     SWPC=17

```

Figure 19. Partial LEXC display

You can also type in exceptions with new order numbers on the command line following the LEXC command. For example:

```

LEXC DNRS=1 WSHI=2 DRDY=3

```

XACB

Sets exception thresholds and attributes.

Type:

Immediate

Format:

```
XACB
  [{ALL|GROUP=cc|LIST=cccc} FORCE]
  [{VERBOSE|TERSE} FORCE]
```

(blank)

Lists all exceptions with their current settings in columnar display. This is the default.

ALL

Lists all exceptions in invocation sequence.

GROUP

Specifies the 2-character identifier (cc) to be associated with the exception.

LIST

Specifies the exceptions (cccc) to display. The exceptions requested are listed with their current settings, which you can modify.

VERBOSE

Multiline display for each defined exception. Displays all exception parameters.

TERSE

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 will appear as if the exception analysis command were actually executing. To display a sample message text of a specific exception, the STATE=TEST must be specified along with the FORCE keyword.

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
```

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

DISPLAY Parameters

STATE

One of the following:

NDSP

Allows you to suppress the display of exceptions that you do not need to act on at this time. OMEGAMON treats the exception as ON, but the exception does not appear. 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 session.

OFF

Does not invoke this exception during the current OMEGAMON session.

TEST

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

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

GROUP

Specifies the 2-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 colors in the following list and follow the same order.

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.

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

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 [Appendix A, "Automating and logging features,"](#) on page 309 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 uses those that you set for the exception.

CYCLE Parameters

EXNCYC

Sets the frequency for checking the exception at every *n* OMEGAMON cycles. If this parameter is set to 0 or 1, it will be tested every OMEGAMON cycle. If it is set to a higher number, it will only be tested each time 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 being examined by an exception (such as DASD or tape), you may 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 appears on the screen as usual. In the following cycles during which it is not scheduled for testing, the exception message redisplay on the screen following 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 will not be tested or displayed during the current OMEGAMON session, unless the user resets this parameter. The (*m*) value, which is informational only, indicates the number of times the exception has already tripped 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 has tripped during the current OMEGAMON session. Users may not alter this value.

Figure 20 on page 99 shows a typical XACB display:

```

XACB
:xcsa
+ DISPLAY Parameters:      THRESHOLD Parameters:      XLF Parameters:
:   State=ON              Threshold=85                Auto=ON   Repeat=YES
:   Group=SP              Display=Red                 Log=BOTH  Persist=25
:   Bell=YES              Attribute=NONE              Limit=0   SS=SETACODE
+ BOX Parameters:         CYCLE Parameters:
:   Boxchar='+'           ExNcyc=0
:   Boxclr=Clr2           Stop=0 (7)
:   Boxattr=BLINK         Cumulative=5
:nvsc
+ DISPLAY Parameters:      THRESHOLD Parameters:      XLF Parameters:
:   State=ON              Threshold=500               NOT ELIGIBLE FOR XLF
:   Group=CT              Display=Yellow
:   Bell=NO               Attribute=NONE
+ BOX Parameters:         CYCLE Parameters:
:   Boxchar=NONE          ExNcyc=3
:   Boxclr=NONE           Stop=0
:   Boxattr=NONE          Cumulative=12

```

Figure 20. Typical XACB display

Figure 21 on page 99 shows 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
: WSL0 Threshold=300 Display=Blue State=NDSP Bell=OFF

```

Figure 21. Partial XACB terse mode display

XTXT

Specifies the message to display when no exceptions have tripped.

Type:

Immediate

Format:

XTXT 'yourmessage'

The variable *yourmessage* can be up to 60 characters of user-defined text, specifying the message to display when no exceptions have tripped. Enclose a message containing blanks or special characters in single quotation marks.

If you enter XTXT alone, OMEGAMON displays the message previously defined for no tripped exceptions.

Summarizing exception activity

The XSUM and XTRP commands show you exception activity.

XSUM

Displays a summary of exceptions and their current status.

Type:

Immediate

Format:

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

(blank)

XSUM with no keywords displays all of the exception groups in alphabetical order. This is the default.

GROUP

The group ID (cc) can be any of the same exception groups used with the XGRP command. Enter a 2-character group ID to summarize the exceptions of one group only.

LIST

The value can be A or I.

A

Lists exceptions in alphabetical order. This is the default.

I

Lists exceptions in invocation sequence.

RESET

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

Figure 22 on page 100 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=HD 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=HD 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=RS 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=RS Limit=3 Persist=2 Auto=OFF Log=NO
+-----+-----+-----+-----+-----+-----+

```

Figure 22. Partial XSUM display

The XSUM command displays the current settings for the exception state (State=ON/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 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)

XTRP with no keywords displays all of the exception groups in alphabetical order. This is the default.

GROUP

The group ID (cc) can be any of the same exception groups used with the XGRP command. Enter a 2-character group ID to display the exceptions of one group only.

LIST

The value can be A or I.

A

Lists exceptions in alphabetical order. This is the default.

I

Lists exceptions in invocation sequence.

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

GDFN

Defines or lists exception groups for exception analysis.

This section lists the commands that allow you to customize your exceptions and save the new definitions in a user profile.

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 appear together on the display.

Type:

Immediate

Format:

```
GDFN
  [GROUP=cc]
  [LIST=cccc,aaaa NAME='ccc|ccc']
  [POSITION=nn]
  [DELETE=EXCEPTION LIST=cccc,aaaa]
  [DELETE=GROUP]
```

(Blank)

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

GROUP=cc

Specifies the 2-character exception group ID. A group ID can be any two unique alphanumeric characters. To list the entries for an existing group, enter this keyword and the group ID.

LIST=cccc,aaaa,||

Lists and adds exceptions to the exception group specified with the GROUP= keyword. An exception may be associated with only one group at a time.

NAME='ccc|ccc'

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

POSITION=nn

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.

DELETE=EXCEPTION

Deletes exceptions specified with LIST= from the group specified by the GROUP= keyword. Note that the syntax does not allow you to use the DELETE keyword followed by a list of exceptions. You must enter the LIST keyword.

DELETE=GROUP

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

For example, to define the group SP and its related exceptions, enter:

```
GDFN GROUP=SP POSITION=1 NAME='SYSTEMS PROGRAMMER'
GDFN GROUP=SP LIST=ABND,ELAP,MAXU,PSCU,SLOG,TSOR,WAIT,XCON
GDFN GROUP=SP LIST=XCSA,XDDR,XECS,XGRS,VISC,XRMF
```

To delete specific exceptions from group TX, enter:

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

To delete the group TX and all its related exceptions, enter:

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

Important: In the examples of the GDFN DELETE keyword, note that you must specify what type of delete function you want GDFN to perform: delete only individual exceptions that you list (DELETE=EXCEPTION *and* LIST=exception,||), or the entire group (DELETE=GROUP).

Controlling groups

The XGSW command gives you control of exceptions by group.

XGSW

Sets exception group switch settings.

The group switch command allows you 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. Overtyping the current setting for the STATE keyword to change the setting.

Type:

Immediate

Format:

XGSW GROUP=cc STATE=cccc

GROUP

Any two unique alphanumeric characters (cc) to specify the group. Use this keyword to display only entries for a particular 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 has been displayed because of TEST mode, a T appears in column 2 of the message lines.)

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

Specifies that the individual exception, rather than the group switch, is to maintain control. This is the default.

Defining groups or mask entries

ASG

Displays or modifies address space threshold groups.

Type:

Immediate

The ASG command lets you display and modify the address space threshold groups table. An address space threshold group is a group of address spaces for which you have defined similar exception analysis characteristics. Displays use the multiline input format so you can easily alter the values. The following considerations apply to the multiline input facility:

- You can modify any display line that begins with a colon in column 1.
- Blanking out fields has no effect; OMEGAMON redisplay the line on the next cycle.
- To change a setting, type over the displayed value and press Enter.
- OMEGAMON marks modified entries with one of the following:

ADDED
UPDATED
DELETED

OMEGAMON treats commas, blanks, and parentheses in command syntax as delimiters.

You can select address space threshold groups by specifying either the entry number of the group within the table or by specifying jobnames in the address space threshold group. This latter capability

helps locate specific entries. The ability to select specific address space threshold groups is also useful, because the display for each address space threshold group is quite long, making it impractical to list several groups at a time.

If you want to define a new address space threshold group, OMEGAMON provides you with a skeleton to fill in after you issue the ASG command. You can also request that your new address space threshold group be modeled after an existing one.

When you enter the ASG command without any additional operands, you see a summary display of the current address space threshold group table; for an example, see [Figure 23 on page 104](#).

```
----- ZWELCOME TSO OM/DEX V420./C SYSA 02/12/09 16:45:31
ASG
+ 1 Jobnames = *MASTER*
+ 2 Jobnames = GRS PCAUTH TRACE CONSOLE ALLOCAS
+ 3 Jobnames = DUMPSRV TSO* RMF*
+ 4 Jobnames = JES*
+ 5 Jobnames = CICS16 CICS17 CICSQM
+ 6 Jobnames = ABCD
+ 7 Jobnames = CICS*
+ There are 7 entries in the address space threshold group table.
```

Figure 23. Summary display of current address space threshold table

Note: OMEGAMON limits the summary display to the first 100 entries.

If you issue the ASG command with the DIS operand, OMEGAMON displays the address space threshold group table in a format that permits modification. The DIS operand permits you to select which address space threshold groups are displayed. The complete syntax is:

```
ASG DIS
      [JOB(cccccccc,cccccccc,||cccccccc)]
      [ENTRY(n1,n2,||nn|n1 : n2)]
      [ALL]
```

where

JOB

Specifies a list of jobnames. You can abbreviate this keyword to any uniquely identifiable string.

ENTRY

Specifies a list, range, or combination of address space threshold group table entries. You can abbreviate this keyword to any uniquely identifiable string.

Note: The threshold group numbers denote the positions of entries in the table only; they have no intrinsic meaning and are not fixed to a particular group. Namely, the entry numbers can be shifted with addition/deletion of entries in the table.

ALL

Specifies all threshold groups.

If you do not specify an operand, the default is DIS. If you use the DIS operand without any keywords, OMEGAMON assumes the ALL keyword. If you use the DIS operand and specify a list or range of numbers, OMEGAMON assumes the ENTRY keyword. For example:

```
ASG DIS 1,2,3
```

is the same as:

```
ASG DIS ENTRY(1,2,3)
```

[Figure 24 on page 105](#) shows typical output from the ASG DIS command.

```

----- ZWELCOME TSO OM/DEX V420./C SYSA 02/12/09 16:45:31
ASG DIS J(*master*,rmf,cics*,clmenu*)
: Address Space Threshold Group 4
: Jobnames = *MASTER*
: Perf Grp =
: SKIPUW = NO
+   THRESHLD CLR BELL           THRESHLD CLR BELL
: ELAP = 2592000 ---- - - - - - WAIT = 10 ---- - - - - -
: NVSC = 500 ---- - - - - - VISC = 500 ---- - - - - -
: WSLO = 4 ---- - - - - - WSHI = 1024 ---- - - - - -
: FXFR = 50 LVL1 ---- - - - - - SWPC = 1 ---- - - - - -
: PSCU = 90 ---- - - - - - TPWT = 60 ---- - - - - -
-----
: Address Space Threshold Group >>> 3 <<<
: Jobnames = DUMPSRV TSO* RMF*
: Perf Grp =
: SKIPUW = NO
+   THRESHLD CLR BELL           THRESHLD CLR BELL
: ELAP = 2592000 ---- - - - - - WAIT = 10 ---- - - - - -
: NVSC = 900 ---- - - - - - VISC = 500 ---- - - - - -
: WSLO = 256 ---- - - - - - WSHI = 1024 ---- - - - - -
: FXFR = 50 RED BELLON ---- - - - - - SWPC = 10 ---- - - - - -
: PSCU = 90 ---- - - - - - TPWT = 60 ---- - - - - -
-----
: Address Space Threshold Group >>> 10 <<<
: Jobnames = CICS16 CICS17 CICSQM
: Perf Grp =
: SKIPUW = No
+   THRESHLD CLR BELL           THRESHLD CLR BELL
: ELAP = 2592000 ---- - - - - - WAIT = 2592000 ---- - - - - -
: NVSC = 500 ---- - - - - - VISC = 500 ---- - - - - -
: WSLO = 4 ---- - - - - - WSHI = 1024 ---- - - - - -
: FXFR = 50 LVL1 ---- - - - - - SWPC = 1000000 ---- - - - - -
: PSCU = 90 ---- - - - - - TPWT = 60 ---- - - - - -
-----
+ OM8400 Entry not found: CLMENU*

```

Figure 24. Output from the ASG DIS command

where

Address Space Threshold Group

Specifies the address space threshold group number.

Jobnames

Specifies the jobnames included in this address space threshold group.

Perf Grp

This field is obsolete (but is still displayed in the command output).

SKIPUW

Specifies whether the WAIT exception should issue a message for this group. Valid values are YES and NO.

THRESHLD

Specifies a threshold for the exception.

ELAP

is in seconds

MVSC

is in 4 k slots

WSLO

is in k

FXFR

is in number of frames

PSCU

is in percent of CPU time

WAIT

is in seconds

VISC

is in 4 k slots

WSHI

is in k

SWPC

is in number of frames

TPWT

is in seconds

CLR

Specifies a color or intensity level for the exception.

BELL

ON or OFF specifies whether the terminal bell should ring when the exception trips.

If you issue the ASG command with the ADD operand, you can add an address space threshold group to the table. The complete syntax is:

```
ASG ADD [ENTRY(n)] [,MODEL(n)]
```

where

ENTRY

Specifies a list or range of address space threshold group table entries. You can abbreviate this keyword to any uniquely identifiable string.

MODEL

Specifies another address space threshold group that you want to use as a model. If you do not use a model, OMEGAMON assigns default threshold values.

If you use the ADD operand without any additional keywords, OMEGAMON adds the address space threshold group to the end of the table. If you specify the number of an existing entry, OMEGAMON inserts the new entry before the existing entry. OMEGAMON deletes the ADD operand and any keywords after you press Enter. [Figure 25 on page 106](#) shows typical output from the ASG ADD command.

```
>ASG 0014
+ Address Space Threshold Group 14 ** ADDED **
: Jobnames =
: Skipuw = NO
+   THRSOLD   CLR   BELL   THRSOLD   CLR   BELL
: ELAP = 2592000  DEFAULT  ----- WAIT = 10  DEFAULT  -----
: NVSC = 500     DEFAULT  ----- VISC = 500  DEFAULT  -----
: WSLO = 4       DEFAULT  ----- WSHI = 1024  DEFAULT  -----
: FXFR = 50      CLR1    ----- SWPC = 1    DEFAULT  -----
: PSCU = 90     DEFAULT  ----- TPWT = 60   DEFAULT  -----
+-----+
+ ENTRY NUMBERS MAY HAVE BEEN SHIFTED BECAUSE OF ADD/DEL
```

Figure 25. Typical output from ASG ADD command

After you add a group to the address space threshold group table, OMEGAMON reorders the group numbers consecutively, if necessary, and displays an informational message telling you that the threshold group numbers may have shifted.

If you issue the ASG command with the DEL operand, you can delete one or more address space threshold groups from the table. The complete syntax is:


```
ASG DEL
inwidth on 'ASG DEL'
[ENTRY(n1,n2,&concat;nn|n1 : n2)]
[ALL]
```

where

ENTRY

Specifies a list, range, or combination of address space threshold group table entries. You can abbreviate this keyword to any uniquely identifiable string.

ALL

Specifies all address space threshold groups in the table.

If you use the DEL operand and specify a list or range of numbers, OMEGAMON assumes the ENTRY keyword. [Figure 26 on page 107](#) shows typical output from the ASG DEL command.

```
>ASG DEL 13
+ Address Space Threshold Group 13 ** DELETED **
+ Jobnames =
+ Skipuw = Yes
+
+ THRSHL D      CLR      BELL      THRSHL D      CLR      BELL
+ ELAP =        0  DEFAULT  -----  WAIT =        0  DEFAULT  -----
+ NVSC =       500  DEFAULT  -----  VISC =       500  DEFAULT  -----
+ WSLO =        64  DEFAULT  -----  WSHI =      1024  DEFAULT  -----
+ FXFR =        25  DEFAULT  -----  SWPC =         2  DEFAULT  -----
+ PSCU =        90  DEFAULT  -----  TPWT =        60  DEFAULT  -----
+-----
```

Figure 26. Typical output from ASG DEL command

After you delete a group from the address space threshold group table, OMEGAMON displays an informational message telling you that the threshold group numbers may have shifted.

CHNM

Displays or sets the channel path mask for missing channel path analysis.

Type:

Immediate

Format:

CHNM [ALL|ONLINE|ADD(nn)|DELETE(nn)]

ALL

Set the channel path mask with all channel paths installed and owned.

ONLINE

Set the channel path mask with all channel paths currently online.

ADD

Add channel path ID *nn* to the existing channel path mask.

DELETE

Delete channel path ID *nn* from the existing channel path mask.

nn

Channel path ID to add to or delete from the mask. You can specify more than one channel path ID if you separate them with blanks or commas.

The default is to display the current channel path mask if you issue the CHNM command without operands.

CHNM displays, sets, or updates the current channel mask, indicating which physical channel paths are operational.

The following figure shows typical output from the CHNM command.

```
CHNM
+      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
+      10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```

The following figure shows typical output from the CHNM ALL command.

```
>CHNM ALL
+      The Channel Path Mask is set to all INSTALLED Channel Paths.
```

The following figure shows typical output from the CHNM ONLINE command.

```
>CHNM ONLINE
+      The Channel Path Mask is set to all ONLINE Channel Paths.
```

The following figure shows typical output from the CHNM ADD command.

```
>CHNM ADD(18)
+      The following Channel Paths were ADDED to the Channel Path Mask.
+      18
```

The following figure shows typical output from the CHNM DELETE command.

```
>CHNM DELETE(02,03)
+      The following Channel Paths were REMOVED from the Channel Path Mask.
+      02 03
```

CPUM

Displays or sets the CPU mask for missing CPU exception analysis.

When the XCPU exception is invoked, it examines the number of active CPUs, and builds a mask of processors that should be active. You can display or reset this mask with the CPUM command.

Type:

Immediate

Format:

```
CPUM [ADD(n) | DELETE(n) | ONLINE]
```

ADD

Specifies a CPU ID to add to the existing CPU mask.

DELETE

Specifies a CPU ID to delete from the existing CPU mask.

ONLINE

Sets the CPU mask to all online CPUs.

You can specify more than one CPU ID on the ADD or DELETE keywords if you separate each with blanks or commas. For example:

```
CPUM ADD(0,2,3)
```

When you change the CPU mask, OMEGAMON returns a message summarizing the changes you made.

You cannot turn off the mask bit for a CPU that is currently active, because XCPU automatically resets it at each cycle. The following figure shows typical output from the CPUM command.

```
CPUM 0 2
```

The following figure shows typical output from the CPUM ADD command.

```
>CPUM ADD(1)
+   The following CPUs were ADDED to the CPU Mask.
+   2
```

The following figure shows typical output from the CPUM DELETE command.

```
>CPUM DELETE(0)
+   The following CPUs were REMOVED from the CPU Mask.
+   1
```

The following figure shows typical output from the CPUM ONLINE command.

```
>CPUM ONLINE
+   The CPU Mask is set to all ONLINE CPUs.
```

MTA

Displays or modifies the missing task table.

Type:

Immediate

The MTA command lets you define the jobs you want included in missing task analysis, a function of exception analysis. It uses the multiline input facility to make it easy for you to change values. The following considerations apply to the multiline input facility:

- You can modify any display line that begins with a colon in column 1.
- Blanking out fields has no effect; OMEGAMON redisplay the line on the next cycle.
- To change a setting, type over the displayed value and press Enter.
- OMEGAMON marks modified entries with one of the following words:

ADDED
UPDATED
DELETED

OMEGAMON treats commas, blanks, and parentheses in command syntax as delimiters.

If you issue the MTA command without any operands or keywords, OMEGAMON displays either the first 100 entries or the entire table. [Figure 27 on page 109](#) shows a typical missing task table.

```
----- ZWELCOME TSO OM/DEX  V420./C SYSA 02/12/09 16:46:01
MTA
+ #  TASK NAME  STATE
: 1  TSO        AUTO
: 2  RMF        AUTO
: 3  JES2       AUTO
: 4  JES3       AUTO
: 5  IMS        AUTO
: 6  CICS       AUTO
: 7  NET        AUTO
: 8  TCAM       AUTO
: 9  VTAM       AUTO
+ There are 9 entries defined in the missing task table
```

Figure 27. Typical missing task table

#

Entry number in the missing task table. Entry numbers are for table display purposes only and cannot be modified.

TASK NAME

Name of the job for which missing task analysis is active.

STATE

State of missing task analysis for the job. Valid values are AUTO or A, ON, and OFF.

AUTO

Warns when the task is missing if it has been active since OMEGAMON startup.

ON

Warns when the task is missing.

OFF

Turns off the missing task analysis for that task.

If you issue the MTA command with the DIS operand, OMEGAMON displays the missing task table in the format shown previously (if you issue the MTA command without operands, OMEGAMON assumes the DIS operand). The syntax is:

```
MTA DIS
```

If you issue the MTA command with the ADD operand, you can add one or more entries to the missing task table. The syntax is:

```
MTA ADD (ccccccc, state)
```

where

ccccccc

Specifies a jobname.

state

Specifies the state of missing task analysis for the job name. Possible values are as follows:

AUTO or A

ON

OFF

Figure 28 on page 110 shows typical output from the MTA ADD command.

```

----- ZWELCOME TSO OM/DEX V420./C SYSA 02/12/09 16:46:01
>MTA ADD (OMEGAMON,ON), (DELTAMON,ON), (DB2,ON)
+ 10 OMEGAMON ON * ADDED *
+ 11 DELTAMON ON * ADDED *
+ 12 DB2 ON * ADDED *
+Entry numbers may have been shifted because of ADD/DEL
+ # TASK NAME STATE
+ 1 TSO AUTO
+ 2 RMF AUTO
+ 3 JES2 AUTO
+ 4 JES3 AUTO
+ 5 IMS AUTO
+ 6 CICS AUTO
+ 7 NET AUTO
+ 8 TCAM AUTO
+ 9 VTAM AUTO
+ 10 OMEGAMON ON
+ 11 DELTAMON ON
+ 12 DB2 ON
+ There are 12 entries defined in the missing task table

```

Figure 28. Typical output from MTA ADD command

If you issue the MTA command with the DEL operand, you can delete one or more entries from the missing task table. The syntax is:

```

MTA DEL
  [cccccccc]
  [ENTRY(n1,n2,||nn|n1 : n2)]
  [ALL]

```

where

cccccccc

Specifies a jobname. You can specify a list of jobnames if you separate each name with commas or blanks.

ENTRY

Specifies a list or range of entries in the table, by number.

ALL

Specifies all entries in the table.

Figure 29 on page 111 shows typical output from the MTA DEL command.

```

----- ZWELCOME TSO OM/DEX V420./C SYSA 02/12/09 16:46:01
>MTA DEL (4,8)
+ #  TASK NAME      STATE
+ 4  JES3           AUTO    * DELETED *
+ 8  TCAM           AUTO    * DELETED *
+Entry numbers may have been shifted because of the ADD/DEL
+ #  TASK NAME      STATE
+ 1  TSO            AUTO
+ 2  RMF            AUTO
+ 3  JES2           AUTO
+ 4  IMS            AUTO
+ 5  CICS           AUTO
+ 6  NET            AUTO
+ 7  VTAM           AUTO
+ 8  OMEGAMON       ON
+ 9  DELTAMON       ON
+ 10 DB2            ON
+ There are 10 entries defined in the missing task table

```

Figure 29. Typical output from MTA DEL command

Setting special thresholds

TPWT

Displays and sets tape mount pending threshold in seconds.

Type:

Immediate

The WAIT exception uses this threshold as a secondary tape mount wait threshold. Before wait analysis considers this threshold, an address space must first exceed the wait exception threshold.

The ASG command also allows specification of the TPWT threshold for an address space group.

Exceptions

There are two basic groups of exceptions: those that apply to the entire system, and those that apply to individual address spaces. The exception descriptions indicate whether the exception is an address space exception.

You can set each of these exceptions to one of the following four states during an OMEGAMON session:

ON

Exception analysis is on for this session.

OFF

Exception analysis is off for this session.

TEST

Show sample warning message during this session.

NDSP

Exception analysis is ON, but is not displayed for this session.

To set exception thresholds and options, refer to the XACB command. (See the ASG command for specific address space exception thresholds.)

ABND (OP)

Indicates address space ABENDING.

Type:

Exception

The ABND exception appears when OMEGAMON detects that a job is abending. This exception will not appear for every job that abends on your system.

Since the ABEND process is generally very quick, it is unusual for an exception to actually catch a job in this state. When this exception trips, it usually means an address space somehow became hung within abnormal termination.

The abend code appears in hex format as it appears in the System Diagnostic Work Area (SDWA).

Bits 0 through 7 are flag bits, bits 8 through 19 are the system abend code, and bits 20 through 31 are the user abend code.

This is an address space exception.

ASER (OP)

Indicates more than *n* total page data set slot errors.

Type:

Exception

ASMI (OP)

Indicates more than *n* outstanding ASM I/Os.

Type:

Exception

These are outstanding requests for page-ins, page-outs, etc.

This value is also known as ASMQ to the SRM.

When an address space is being swapped in or out of memory, a large number of I/O requests can be dumped into the queue at the same time; this may cause ASMQ to spike for a brief period before the requests are serviced. As a result, the ASMI exception may occasionally warn of a large number of waiting requests. You can ignore this as long as the queue is not extremely large or does not last for a long time.

DNRS (HD)

Indicates device not responding.

Type:

Exception

The DNRS exception produces a message if an active I/O issued to any DASD allocated to z/OS takes longer than one OMEGAMON cycle to complete (default is five seconds).

The delay can be due, for example, to a head-of-string contention, hardware errors, or to a device that has dropped ready. As long as this I/O fails to complete, the message appears and device response time increases. When the I/O finally completes, the message disappears.

DRDY (HD)

Indicates DASD dropped ready.

Type:

Exception

A DASD device that has dropped ready can halt all batch, online systems, TSO, and even the entire operating system in the case of a JES2 or PLPA page data set.

This exception does *not* appear if the device is simply processing a mount, or if there are no I/O requests pending against it.

DYPR (HD)

Indicates 3380-type devices not using Dynamic Path Reconnect.

Type:

Exception

Note: If you run z/OS as a guest under VM, this exception always appears, so it is best to turn it off.

ELAP (OP)

Indicates address space elapsed time is greater than *n* seconds.

Type:

Exception

This is an address space exception.

ELAP is not applicable to STC address spaces.

FXFR (OP)

Indicates address space using more than *n* fixed frames.

Type:

Exception

This is an address space exception.

MAXU (OP)

Indicates more than *nn*% of ASVT slots in use and unusable.

Type:

Exception

The MAXU exception warns when the percentage of in-use and unusable slots exceeds the threshold, and reports the percentage of Reserve that has been used.

The ASVT minor of SYS indicates the status of all ASVT slots.

NVSC (OP)

Indicates more than *n* non-VIO slots in use.

Type:

Exception

This is an address space exception.

PAER (OP)

Indicates more than *n* errors on a page data set.

Type:

Exception

The PAER exception produces a warning when the number of local slot errors on any page data set exceeds the threshold.

Individual page data set errors are accumulated by the Auxiliary Storage Manager (ASM) for all page data sets.

PAIO (OP)

Indicates paging I/O request more than *nnn* milliseconds old.

Type:

Exception

It is not unusual for paging I/O to be occasionally degraded, but if it happens frequently, you should investigate the paging volumes involved.

PAOD (OP)

Indicates paging volume has *n* open DCBs.

Type:

Exception

The PAOD exception warns when a paging volume has open data sets (DCBs). It also reports the volser and unit address of the device where the page data set resides.

PATM (OP)

Indicates average page I/O time more than *nnn* milliseconds.

Type:

Exception

The PATM exception produces a warning when the service burst size in milliseconds for any page data set exceeds the threshold.

This value is a reflection of how long it takes on the average to read or write a page to the page data set. Although this value relates to measures such as the MilliSeconds Per Page (MSPP) and the device service time, it is not directly comparable to any of them; it is a somewhat artificial value that the ASM uses internally.

PDNO (OP)

Indicates page data set not operational.

Type:

Exception

PERA (OP)

Indicates Program Event Recording is active for specific address spaces.

Type:

Exception

PERS (OP)

Indicates Program Event Recording is active for all address spaces on the system.

Type:

Exception

PSCU (OP)

Indicates more than *nn%* of step CPU limit used.

Type:

Exception

The PSCU exception warns when any address space uses up more than *nn%* of the CPU time allowed in the current job step (where *nn* is the threshold for this address space).

If a batch job, for example, exceeds its threshold of CPU time, it could be in imminent danger of terminating with a system 322 abend.

This is an address space exception.

SDNO (OP)

Indicates swap data set not operational.

Type:

Exception

SLOG (OP)

Indicates SYSLOG not recording.

Type:

Exception

SLOT (OP)

Indicates fewer than *n* slots available.

Type:

Exception

The SLOT exception warns when the total number of available ASM slots falls below the threshold.

This means that you need to either define more page data sets, or cut back on the number of active address spaces.

SSRT (OP)

Indicates I/O time for swap data set more than *nnn* milliseconds.

Type:

Exception

Format:

I/O SERVICE TIME =*ttt* MS ON SWAP DS *ddddddd* (*vvvvvv/cuu*)

ttt

The I/O time in milliseconds.

ddddddd

The name of the swap data set.

vvvvvv

The volume serial identification of the device the swap data set is on.

cuu

The address of the device the swap data set is on.

RMF must be active for SSRT to work.

SWER (OP)

Indicates more than *n* errors on a swap data set.

Type:

Exception

The SWER exception warns when the number of slot errors on any swap data set exceeds the threshold.

Individual swap data set errors are accumulated by the Auxiliary Storage Manager (ASM) for all swap data sets.

SWPC (OP)

Indicates address space has swapped more than *n* times.

Type:

Exception

A job's swap count includes both physical and logical swaps.

This is an address space exception.

TNRS (OP)

Indicates tape not responding.

Type:

Exception

The TNRS exception warns if an I/O issued to any tape device allocated to a z/OS address space takes longer than 1 OMEGAMON cycle to complete (default five seconds).

As long as this I/O fails to complete, the message appears and the tape response time increases. When the I/O finally completes, the message disappears.

TNRS also reports a problem if z/OS tries to issue an I/O to a device that has dropped ready.

TRDY (HD)

Indicates tape device dropped ready.

Type:

Exception

If any I/O was in progress on this device at the time of failure, the TNRS exception warns an additional warning.

VISC (OP)

Indicates more than n VIO slots in use.

Type:

Exception

This is an address space exception.

VTOC

Displays unit address and volser of indexed VTOC switched to OS VTOC.

Type:

Exception

This exception warns when the system switches an indexed VTOC to an OS VTOC.

The VTOC exception displays the unit address and volser for all packs that had indexed VTOCs when OMEGAMON was initialized, but that no longer appear to have indexed VTOCs. The exception appears until the volume is taken offline, or the OS VTOC is switched back to an indexed VTOC.

This command requires the Common VTOC Access Facility (CVAF).

Note: This exception does not check either offline volumes or volumes for which the CVAF cannot determine whether they contain an indexed VTOC.

WAIT (OP)

Indicates address space wait time longer than n seconds.

Type:

Exception

The WAIT exception warns when any address space waits longer than the threshold defined by the installation.

A WAIT is incurred when no TCB time is used, even if SRB time is used. After determining that the address space is indeed hung up, the WAIT exception tries to discover why the address space is waiting. Only address spaces with active transactions are examined; any idle initiators or TSO users waiting on terminal input are ignored.

This is an address space exception.

With the WAIT exception, the words Wait or Swap appear, depending upon whether the address space is resident in virtual memory (swapped in).

The time the address space is waiting or swapped appears as:

59 Sec

for 59 seconds

1:01 Mn

for 61 seconds (1 minute, 1 second)

1:01 Hr

for 61 minutes (1 hour, 1 minute)

1:01 Dy

for 25 hours (1 day, 1 hour)

After it establishes that a job has been waiting unusually long, OMEGAMON then analyzes *why* the address space is waiting. The following topics describe these swap and wait reasons.

Swap reasons

When an address space is not currently in real memory, a swap reason appears. The Exchange and Unilateral swaps are also flagged as Swapped by the SRM; this is because these swap reasons are due to an SRM decision to adjust the system Multi-Programming Level. For a detailed description of these swap types, refer to the *IBM MVS Initialization and Tuning Guide*.

The swap reason might be one of the following:

APPC-Wt

APPC wait swap

Aux-Strg

Auxiliary storage swap

Cent-Str

Central storage

Det-Wait

Detected wait swap

Enq-Xchg

Enqueue exchange swap

Exchange

Exchange on recommendation

Long-Wat

Long wait swap

Real-Str

Real storage swap

Request

Request swap

Sys-Page

System paging swap

Term-In

Input terminal wait swap

Term-Out

Output terminal wait swap

Too-Long

Out too long

Transwap

Transition swap

Unilatrl

Unilateral swap

Enqueue waits

If an address space tries to enqueue a resource that is already owned by another user, an enqueue conflict occurs and the requester must wait. Resources are defined by a 1- to 8-character major name and a variable-length minor name. For example, when the system enqueues a data set, it always uses SYSDSN for the major name, and the data set name itself for the minor name.

- Data sets Held by *aaaaaaaa* GRS SYSNAME: *ssssssss*

Most enqueue waits are for data set enqueues. The second line, which displays the GRS System ID of the current resource owner only appears if the owner is on a different system from the one waiting.

- Waiting on Job: *aaaaaaaa* (Exc) EXC ENQ: *pppppppp qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq*

For any enqueue wait *other* than for a data set, this form of the exception message displays both the major (*pppppppp*) and minor (*qqqq&concat;*) names of the resource.

- Waiting in Allocation for: *aaaaaaaa*

Waiting in allocation is a special type of enqueue wait. When a job is in the middle of device allocation, it must acquire the SYSIEFSD/Q4 enqueue to serialize access to the allocation process. If another user is already in the middle of allocation, then other jobs must wait.

- Waiting in Allocation Waiting on Device Swap for: *aaaaaaaa Spool Full! Current User: aaaaaaaaa*

If the job is the *owner* of the SYSIEFSD/Q4 enqueue, these messages may appear because other jobs are waiting for it. This condition usually results from an unanswered REPLY DEVICE, WAIT, OR CANCEL message on the z/OS operator's console.

Device waits

It is very common for an address space to be delayed while waiting for I/O to some device to complete.

The following are possible device wait reasons:

- DASD not ready
- Control unit busy
- Channel busy
- CHN/DEV end pending
- Device busy
- Mass Storage System (MSS) staging
- DASD reserve
- Tape mount
- Tape rewinding

Note: WAIT analysis uses the additional TPWT threshold in determining excessive tape mount pending time. Both the WAIT threshold and TPWT threshold must be exceeded for this condition.

Miscellaneous waits

Following are examples of various miscellaneous wait reasons that may appear:

- Cross Memory Post SRB Pending
- Swapping In
- Job Select Delayed: No Pageable Frames
- Job Select Delayed: ASM Slot Shortage
- DPpty = 0 Sequence # = *nn*
- Local Lock Held: Dispatchable
- Local Lock Held: Non-Dispatchable
- Local lock held: CPU *n*
- *n* Suspended SRB(s) Waiting for Dispatch
- Waiting for Paging

Unresolved waits

In some cases, it is impossible for OMEGAMON to discover the actual wait reason. This can be because the job is in an unanalyzable z/OS state, or simply because the job has gone into an internal wait on one of its own event control blocks (ECBs). The second reason is quite common for started tasks such as RMF, because they often put themselves to sleep until some event occurs.

This type of wait is not a real exception, because it is considered part of the normal operation of the address space. The WAIT exception still produces a warning message for these jobs, unless you assigned them to an address space threshold group with the SKIPUW=YES option specified. This option tells

OMEGAMON to skip the warning message if the wait is unresolved. The ASG DIS ENTRY(*nn*) immediate command displays the thresholds for each address space threshold group *nn*.

WTOR

Indicates number of available WTOR buffers is $< n$.

Type:

Exception

n is the threshold value specified for the WTOR exception. The default is 5.

WSHI (RS)

Indicates address space working set $> n K$.

Type:

Exception

The working set size is calculated as four times the frame count of a swapped in address space (displayed by the FMCT command).

The Exception Logging Facility (XLF) can process this address space exception.

WSLO (OP)

Indicates address space working set $< n K$.

Type:

Exception

This is an address space exception.

The working set size is calculated as four times the frame count of a swapped in address space (displayed by the FMCT command).

XACP (OP)

Indicates average CPU utilization $> nn\%$.

Type:

Exception

This threshold is *not* the happy value, although your installation may choose to set them at the same value.

The average CPU utilization value calculated by the SRM varies from 0 to 100 percent, even if there are two or more processors active; that is, the value is a utilization for the system as a whole. This value can actually be set by the SRM to 128 percent. This is a signal that, during the previous SRM interval, none of the processors ever entered the wait state, and more work (up to 28 tasks) is waiting for the CPU.

In an LPAR environment, SRM does not take into account LPAR non-dispatch time, which can lead to inflated values being reported. Use the XACR exception instead.

You can display the current average CPU utilization by using the RCTA, RCPD, or RCPD minor commands of SYS.

XACR (HD)

Indicates average CPU utilization, as computed by RMF, exceeds *nn*%.

Type:

Exception

To use this exception, RMF API support must be installed. In addition, the Monitor I CPU report must be active. You can display the current average cpu utilization as computed by RMF using the RCPR minor command of SYS.

XCHN (HD)

Indicates a missing channel path.

Type:

Exception

The XCHN exception warns if a channel path unexpectedly goes inoperative or is offline. The XCHN exception does not warn if the logical path to devices goes offline.

The CHNM command displays or changes a channel path mask that tells XCHN which channel paths should normally be operational.

The XCHN exception warns when channel *nn* is not operational, where *nn* is the Channel Path ID.

XCON (OP)

Indicates fewer than *n* WTO buffers available.

Type:

Exception

If the supply of Write to Operator (WTO) buffers is depleted, the z/OS master console may become locked out; this sometimes occurs if a console has been made non-rollable. The XCON exception warns when the number of available z/OS console buffers falls below a certain threshold.

A shortage of console buffers is often the result of a hardware failure at a hardcopy log printer; output destined for the printer backs up, and eventually uses up all the available console buffers. In this case, the printer must be quickly varied offline before the z/OS consoles themselves become locked out.

Because OMEGAMON does not use WTO/WTOR to communicate, it may continue to display messages even after other address spaces are unable to communicate to the operator via WTO.

XCPU (HD)

Indicates a missing CPU.

Type:

Exception

The XCPU exception warns if a CPU is lost.

The CPUM command displays or changes the CPU mask to tell XCPU which processors are expected to be online.

XCSA (OP)

Indicates common Storage Area (CSA) utilization > *nn*%.

Type:

Exception

The Common Service Area (CSA) has a fixed amount of virtual memory available to it, based on the CSA parameter in the IEASYSxx member of PARMLIB (and any SQA overflow).

The threshold at which this exception message becomes critical depends on the type of workload that runs on your system. For example, IMS systems, whose workload is fairly constant, often run with a very high percentage of allocated CSA. However, if your system has a mixed workload, which grows during the day, a high percentage of allocated CSA may become a problem.

XDDR (OP)

Indicates DDR swap in progress.

Type:

Exception

The XDDR exception indicates that a Dynamic Device Reconfiguration (DDR) swap is in progress.

This event can drastically affect the performance of z/OS, especially by causing erratic TSO response time.

This exception shows the from and to device addresses, which can indicate:

- Direct access device address
- Magnetic tape device address
- Unit record device address

XECS (OP)

Indicates extended CSA > *nn*% of maximum available.

Type:

Exception

XGRS (OP)

Indicates Global Resource Serialization (GRS) ring has broken.

Type:

Exception

The XGRS exception warns if the GRS ring breaks.

This exception only has significance for those installations using GRS in a multiple system environment, where various systems are exchanging resource serialization information with the GRS ring of channel-to-channel adapters.

XGTF (OP)

Indicates Generalized Trace Facility (GTF) is active.

Type:

Exception

The XGTF exception indicates that the GTF is active.

GTF is an optional service program that records significant system events, such as supervisor calls and start I/O operations. This event can drastically affect the performance of z/OS, especially by causing erratic TSO response time.

XJCA (OP)

Indicates if any job's growth in usage is greater than *nn*% of the maximum size of ECSA (requires the CSA Analyzer).

Type:

Exception

XJCB (OP)

Indicates if any job's growth in usage is greater than *nn*% of the maximum size of CSA (requires the CSA Analyzer).

Type:

Exception

XJSA (OP)

Indicates if any job's growth in usage is greater than *nn*% of the maximum size of ESQA (requires the CSA Analyzer).

Type:

Exception

XJSB (OP)

Indicates if any job's growth in usage is greater than *nn*% of the maximum size of SQA (requires the CSA Analyzer).

Type:

Exception

XMCA (OP)

Indicates Monitor Call is active on the system.

Type:

Exception

XMEM (HD)

Indicates real storage online is less than *n* K.

Type:

Exception

The XMEM exception warns if the amount of currently available real storage (in K) is less than the threshold.

If the actual amount of real storage is *greater* than the threshold, the threshold automatically sets equal to the current value. This means that if you *underestimate* the true amount of real storage on your system, OMEGAMON sets the current value for you.

When you set this threshold, do not take into account the size of the HSA. During initialization, OMEGAMON automatically decreases this threshold by the size of the HSA.

XMTA (OP)

Indicates a missing address space.

Type:

Exception

In every z/OS system some address spaces are considered critical and must be active at all times, such as JES2 (or JES3), VTAM, RMF, CICS®, and IMS. The XMTA exception warns if any of these critical address spaces disappears from the system. The MTA command can customize the list of critical address spaces.

You can set the exception for each address space to one of the following states:

ON

Warn when this task is missing.

OFF

Do not warn when this task is missing.

AUTO

Warn when this task is missing only if it has been active since IPL, but is no longer.

You can use the MTA command to display and modify the missing task table interactively. If *nn* = 0, MTA displays the entire missing address space table; a non-zero value changes the *n*th entry within the table by typing over the display.

XOLT (OP)

Indicates On Line Test Executive Program (OLTEP) is active.

Type:

Exception

OLTEP is a facility that schedules and controls activities on the On Line Test System (OLTS), and provides communications with the operator.

If OLTEP is active, it can drastically affect the performance of z/OS subsystems such as TSO by causing, for example, erratic TSO response time.

XPRT (SR)

Indicates the average total page fault rate threshold > *n*.

Type:

Exception

This threshold is *not* the happy value, although your installation may choose to set it at the same value.

To display the current average total page fault rate, use the RPAG minor command of SYS.

XREP (OP)

Indicates more than *n* outstanding operator replies.

Type:

Exception

The XREP exception warns when the number of outstanding z/OS operator replies is greater than the threshold.

In many installations, it is common for some subsystems to continually have an outstanding reply, for example, operator utilities.

Some installations expect to have a certain number of WTOR messages on the consoles at all times, because certain subsystems (such as IMS) use this as a mechanism to accept subsystem commands. Each installation, therefore, normally has a different number of outstanding WTORS.

XRMF (OP)

Indicates RMF Monitor I not active.

Type:

Exception

This command requires RMF.

The XRMF exception indicates if the RMF address space is active, but the Monitor I subtask (ZZ) is not.

XSCA (OP)

Indicates the system's growth in usage is greater than $nn\%$ of the maximum size of ECSA (requires the CSA Analyzer).

Type:

Exception

XSCB (OP)

Indicates the system's growth in usage is greater than $nn\%$ of the maximum size of CSA (requires the CSA Analyzer).

Type:

Exception

XSMF (OP)

Indicates System Management Facilities (SMF) problems.

Type:

Exception

Most installations use SMF for accounting and performance reporting purposes.

The XSMF exception warns of problems with SMF which have made it unable to continue recording data. The most common of these is when all SMF data sets are allowed to fill up.

XSQA (OP)

Indicates SQA and ESQA overflows more than n K into CSA/ECSA.

Type:

Exception

Use SYS minor commands SQA, CSA, ESQA, and ECSA to investigate the XSQA exception. These commands show the exact amounts of SQA overflow into CSA, ESQA overflow into ECSA, and the total size of each.

XSSA (OP)

Indicates the system's growth in usage is greater than $nn\%$ of the maximum size of ESQA (requires the CSA Analyzer).

Type:

Exception

XSSB (OP)

Indicates the system's growth in usage is greater than $nn\%$ of the maximum size of SQA (requires the CSA Analyzer).

Type:

Exception

XUCA (OP)

Indicates if any job's usage is greater than $nn\%$ of the maximum size of ECSA (requires the CSA Analyzer).

Type:
Exception

XUCB (OP)

Indicates when any job's usage is greater than $nn\%$ of the maximum size of CSA (requires the CSA Analyzer).

Type:
Exception

XUIC (SR)

Indicates average unreferenced interval count less than n .

Type:
Exception

XUSA (OP)

Indicates if any job's usage is greater than $nn\%$ of the maximum size of ESQA (requires the CSA Analyzer).

Type:
Exception

XUSB (OP)

Indicates if any job's usage is greater than $nn\%$ of the maximum size of SQA (requires the CSA Analyzer).

Type:
Exception

Chapter 7. General system information

System commands display information about system-wide utilization.

The following are the areas of analysis:

System information

General system information (SYS command and its minor commands).

CPU utilization

Plot of processor utilization by user and total system (MCPUnn, SCPUUnn, SIFAnn, and SIIPnn commands).

Paging

Plot of page-in and page-out rates by user and total system (MPGIInn and MPAGnn commands).

Page data sets

Page data set statistics such as type, size, and status (PART command and its minor commands).

Enqueues

System-wide view of enqueue conflicts.

Data sets

Volume serial identification where data set resides, users of data set.

Formatted control block dumps

Dumps of critical control blocks within z/OS.

LPA Modules

Status, entry point address, current use count.

GRS

Global resource serialization status.

System information

The system information major SYS command and its minor and immediate commands provide general system information.

System information major command

SYS

Displays general system information.

Type:

Major

The SYS major command displays the two-character identifier appended to the following system PARMLIB member:

IEAOPTcc

Contains the parameters that affect swapping and other decisions made by the SRM.

SYS also shows the SYSRES volume and address, and whether there is an active page or swap data set.

```
SYS      >> WLM Goal mode OPT=00 SYSRES=(G1A01F,824A) <<
         The SYSRES volume has an active page or swap data set
```

SYS has minor commands that provide information on many aspects of your system.

System information minor commands

SYS has many minor commands, which are divided into the following groups:

- [“Paging” on page 126](#)
- [“System storage” on page 127](#)
- [“Frame Information” on page 133](#)
- [“Hiperspace commands” on page 135](#)
- [“Swap count commands” on page 136](#)
- [“SRM information” on page 137](#)
- [“Miscellaneous system information minor commands” on page 140](#)

Paging

The SPAL, SPC, SPCI, SPS, SPSV, and SPWS minor commands display paging information.

Paging numbers are stored in a wraparound field within z/OS, so the numbers may be meaningless. It is, therefore, better to append .R to these commands to show the value as a rate of pages/second. A symbol in column 1 becomes the plot symbol. The GSCL command sets the scale of the plot.

In the following example, 2.0/pt indicates that each point on the scale represents two pages/second. The dollar sign (\$) is the plot symbol.

```
SYS    >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
$SPR .R 90.0      2.0/pt  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$SPS .R 44.0      2.0/pt  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$SPWS.R 2.0       2.0/pt  $
```

Figure 30. Use of plot symbol

SPAL

Displays total paging per second.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SPC

Displays common area paging.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SPCI

Displays common area page-ins.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SPS

Displays system area paging.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SPSV

Displays system VIO paging.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SPWS

Displays swap paging.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

System storage

The system storage minor commands of SYS display virtual and real storage.

The commands have the following usual protect key assignments:

1

Job Scheduler and JES

2,3,4

Reserved

5

Data Management: IOS, OPEN/CLOSE/EOV

6

TCAM and VTAM

7

IMS & DB2[®]

8

All V=V programs

9-15

V=R programs

CSA

Displays common storage area (CSA) by subpool and protect key.

CSA displays the number of bytes used as well as allocated for those subpools that are not in fetch-protected memory.

Protected subpools display *AUTH* in the Storage Used column. OMEGAMON cannot access these protected subpools.

The display of CSA utilization shows the extents, allocated blocks, storage used, and the largest free block (see [Figure 31 on page 127](#)).

```
SYS >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
CSA Subpl-Key Ext Alloc. Blks Storage Used Largest Free Blk
+ 227-6 4 16K *AUTH* *AUTH*
+ 231-6 13 272K 267K+384 3K+832
+ 241-6 9 48K 47K+784 240
+ 228-1 1 4K 56 3K+968
+ 231-1 2 20K 16K+680 3K
+ 241-1 8 32K 30K+96 1K+16
+ 241-0 3 12K 11K+96 326
+ 227-0 3 20K *AUTH* *AUTH*
+ 231-0 16 76K 65K+1016 5K+56
+ -----
+ Total: 59 500K 439K+40 17K+390
+ Percent of CSA: 16.6% 14.6% 0.6%
+ CSA Size 3012K
display
```

Figure 31. Output of CSA command

where

Subpl-Key

Storage by subpool and key.

Ext

Number of extents.

Alloc. Blks

Storage that the VSM assigned to the subpool.

Storage Used

Part of the subpool allocation that has been used to satisfy a GETMAIN request.

Largest Free Blk

Largest piece of storage that is in a subpool allocation, but has not been used to satisfy a GETMAIN request.

Total

Ext, Alloc. Blks, and Storage Used totals are sums of the subpools displayed. Largest Free Blk is the total free blocks of all subpools or of unallocated storage.

When an SQA overflow condition occurs, the following line appears at the bottom of CSA output:

```
(CSA SIZE DOES NOT INCLUDE nnnk OF CSA CONVERTED TO SQA)
```

To display CSA for real storage, use the CSAR minor command of SYS.

CSAR

Displays the users of Common Storage Area (CSA) real storage by storage protect key.

Figure 32 on page 128 shows a typical CSAR display.

```
SYS  >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
CSAR  Key   Used by   CSA Real Memory
+     0   Supervisor  592K
+     1     JES       172K
+     5   Data Mgmt.    8K
+     6   TCAM/VTAM   588K
+     7     IMS      268K
+     8     V=V       8K
+     -----
+     Total          1636K
```

Figure 32. Typical CSAR output

ECSA

Displays extended common storage area (ECSA) by subpool and protect key.

Figure 33 on page 128 shows a typical display of ECSA utilization.

```
SYS  >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
ECSA Subpl-Key Ext   Alloc. Blks   Storage Used   Largest Free Blk
+    227-0     1     4K           72           3K+952
+    228-5     1    24K          24K           1K+264
+    241-0     1     4K          2K+72          3K+480
+    241-1     1     8K          4K+544
+    -----
+   Total:      4    40K          30K+688          960K
+   Percent of ECSA:  4.0%          3.1%          96.0%
+   ECSA size:    1,000K
```

Figure 33. Typical ECSA output

where

Subpl-Key

Storage by subpool and key.

Ext

Number of extents.

Alloc. Blks

Storage that the VSM assigned to the subpool.

Storage Used

Part of the subpool allocation that has been used to satisfy a GETMAIN request.

Largest Free Blk

Largest piece of storage that is in a subpool allocation, but has not been used to satisfy a GETMAIN request.

Total

Ext, Alloc. Blks, and Storage Used totals are sums of the subpools displayed. Largest Free Blk is the total free blocks of all subpools or of unallocated storage.

When an ECSA overflow condition occurs, the following line appears at the bottom of ECSA output:

```
(ECSA SIZE DOES NOT INCLUDE nnnk OF ECSA CONVERTED TO ESQA)
```

ESQA

Displays ESQA by subpool and protect key.

Figure 34 on page 129 shows a typical display of ESQA utilization.

```

SYS    >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
ESQA  Subpl-Key  Ext      Alloc. Blks      Storage Used      Largest Free Blk
+      239-0      6          84K              83K+720           3K+928
+      245-0      7          8M+108K          700K+232          7M+928K+440
+
+      Total:      13          8M+192K          783K+952
+      Percent of ESQA:      100.0%          9.4%
+      ESQA size:      8M+192K
=====

```

Figure 34. Typical ESQA output

where

Subpl-Key

Storage by subpool and key.

Ext

Number of extents.

Alloc. Blks

Storage that the VSM assigned to the subpool.

Storage Used

Part of the subpool allocation that has been used to satisfy a GETMAIN request.

Largest Free Blk

Largest piece of storage that is in a subpool allocation, but has not been used to satisfy a GETMAIN request.

Total

Ext, Alloc. Blks, and Storage Used totals are sums of the subpools displayed. Largest Free Blk is the total free blocks of all subpools or of unallocated storage.

When an ESQA overflow condition occurs, the following line appears at the bottom of ESQA output:

```
(ESQA SIZE DOES NOT INCLUDE nnnk OF ECSA CONVERTED TO ESQA)
```

RMAP

Displays map of real memory.

RMAP shows a map of system virtual storage, showing how much central storage (in bytes) is currently assigned to each area.

After RMAP lists all of the virtual storage backed by real frames, it lists any and all queues of real storage frames not associated with virtual storage, if there is at least one item in the queue.

The RSM internal frame queues list below the PSA display line. To list, a queue must have something in it. These queues are:

Available

Frames on the pageable frame queue and marked as available (not having paging I/O in progress, etc.).

Data Spaces

Real storage in use for data spaces.

Data Space Mgmt

Real storage in use for data space management.

BDF

Bottom Double Frames: real storage frames used for segment table entries.

TDF

Top Double Frames: real storage frames used for segment table entries.

SQA Reserved

Frames reserved by the system, which can be used as SQA in an emergency. Because these frames are reserved for emergencies, they are unavailable for other use.

HSA

Hardware System Area. This area is reserved at IMPL time. OMEGAMON displays only the amounts detected by the operating system. Only the operating system can use this storage.

SDUMP

Real storage buffer frames.

DFF

Deferred FREEMAIN frames.

DAT Off Nucleus

Nucleus frames for which Dynamic Address Translation is off.

IDA

System software frames.

Flawed

Unusable frames.

Uninitialized

Frames not used (not initialized).

Unqueued

Frames not used (not queued to anything).

[Figure 35 on page 131](#) shows a typical display for the RMAP command.


```

SYS  >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
RMAP MAJOR AREA          REAL  MINOR AREA          NOT/FIX    FIXED    TOTAL
=====
+   EXTENDED PRIVATE    8,704K                                4,444K    4,260K    8,704K
+                                     (ELSQA)                                20K      4,108K    4,128K
+   -----
+   EXTENDED COMMON    7,696K    CSA          1,156K      416K    1,572K
+                                     PLPA         1,372K      236K    1,608K
+                                     SQA                                     2,864K    2,864K
+                                     READ/WRITE NUC 548K      548K
+                                     READ-ONLY NUC  1,104K    1,104K
+   -----
+   COMMON              5,232K    READ-ONLY NUC 120K      120K
+                                     READ/WRITE NUC 168K      168K
+                                     SQA           496K      496K
+                                     PLPA          2,544K    12K      2,556K
+                                     MLPA          4K        4K
+                                     CSA           1,288K    600K    1,888K
+   -----
+   PRIVATE             26,072K    V=V          20,340K    5,732K    26,072K
+                                     (LSQA)                                2,596K    2,596K
+   -----
+   ABS ZERO FRAME      4K                                     4K        4K
+   =====
+   AVAILABLE           908K
+   DATA SPACES       100K
+   DATA SPACE MGMT    20K
+   BDF                 32K
+   TDF                 32K
+   SQA RESERVED        8K
+   HSA                 448K
+   DAT OFF NUCLEUS     8K
+   OFFLINE             8K
+   -----
+   TOTAL FRAMES        48M
+   =====

```

Figure 35. Typical output of RMAP command

RMLF

Displays real memory statistics for large frames and large pages.

The following statistics are displayed:

- Number of large memory objects allocated in the system
- Number of large pages backed in real storage
- Reconfigurable large frame area, in megabytes
- Nonreconfigurable large frame area, in megabytes
- Number of 1 Mb frames in the reconfigurable large frame area that are allocated
- Number of 1 Mb frames in the nonreconfigurable large frame area that are allocated
- Threshold number of available large frames needed to satisfy nonpreferred requests from the large frame area
- Number of available large frames groups
- Number of available single large frames
- High watermark of the number of large pages allocated on behalf of large page requests
- High watermark of the number of large pages allocated on behalf of 4 K page requests

Figure 36 on page 132 shows a typical display for the RMLF command.

```

----- ZIN      VTM      OM/DEX * V510./C SYS  12/19/11 17:47:30  22
=====
PEEK  M5D0M2RC ASID=371, collected at 17:46:41
lpag
+     Large Memory Objects                0
+     Large Pages Backed in Real          0

SYS   >> WLM Goal mode OPT=GG SYSRES=(G1B01B,8350) <<
rmlf  Large Frame Statistics
+     Large Memory Objects                2
+     Large Pages Backed In Real          128
+     Recon Large Frame Area In Mb        0
+     Nonrecon Large Frame Area In Mb     1024
+     Recon Large Frame Used In Mb        0
+     Nonrecon Large Frame Used In Mb     1023
+     Large Frame Threshold                0
+     Large Frame Groups                  1
+     Available Single Large Frames       114221
+     HWM Large Page 1m Requests          128
+     HWM Large Page 4k Requests          896

```

Figure 36. Typical output of RMLF command

SQA

Displays SQA by subpool and protect key.

SQA displays the number of bytes used as well as allocated for those subpools that are not in fetch-protected memory.

Protected subpools display *AUTH* in the Used column. This is because the control block OMEGAMON needs to get the information, the FQE, is in the subpool.

The SQA utilization display shows the extents, allocated blocks, storage used, and the largest free block.

```

SSYS   >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
SQA   Subpl-Key  Ext      Alloc. Blks      Storage Used      Largest Free Blk
+     245-0      1      384K          221K+16          160K
+
+     Total:      1      384K          221K+16
+     Percent of SQA:      100.0%      57.6%
+     SQA Size      3012K

```

Figure 37. Typical output for SQA command

where

Subpl-Key

Storage by subpool and key.

Ext

Number of extents.

Alloc. Blks

Storage that the VSM assigned to the subpool.

Storage Used

Part of the subpool allocation that has been used to satisfy a GETMAIN request.

Largest Free Blk

Largest piece of storage that is in a subpool allocation, but has not been used to satisfy a GETMAIN request.

Total

Ext, Alloc. Blks, and Storage Used totals are sums of the subpools displayed. Largest Free Blk is the total free blocks of all subpools or of unallocated storage.

When an SQA overflow condition occurs, the following line is appended to the output from the command:

```
(SQA SIZE DOES NOT INCLUDE nnnk OF CSA CONVERTED TO SQA)
```

VMAP

Displays map of z/OS virtual storage.

VMAP shows how big each area is and what virtual range defines it.

```

SYS  >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
VMAP MAJOR AREA          SIZE  MINOR AREA          SIZE  ADDRESS RANGE
+-----+-----+-----+-----+-----+-----+
+ EXTENDED PRIVATE  2,017M          2,017M  01F00000-7FFFFFFF
+-----+-----+-----+-----+-----+-----+
+ EXTENDED COMMON   15M  CSA          3,008K  01071000-01EFFFFF
+                               PLPA         2,252K  019DD000-01C0FFFF
+                               SQA          8,448K  0119D000-019DCFFF
+                               READ/WRITE NUC  548K   01114000-0119CFFF
+                               READ-ONLY NUC  1,101K  01000000-01113667
+-----+-----+-----+-----+-----+-----+
+ COMMON            6M  READ-ONLY NUC  120K   00FE2000-00FFFFFF
+                               READ/WRITE NUC 168,128 00FB8000-00F10BFF
+                               SQA          512K   00F38000-00FB7FFF
+                               PLPA         4,004K  00B4F000-00F37FFF
+                               MLPA         4K     00B42000-00B42FFF
+                               CSA          3,384K  00800000-00B4DFFF
+-----+-----+-----+-----+-----+-----+
+ PRIVATE           10,236K  V=V          8,172K  00005000-007FFFFF
+                               (V=R)         512K   00005000-00084FFF
+                               SYSTEM AREA  16K    00001000-00004FFF
+-----+-----+-----+-----+-----+-----+
+ ABS ZERO FRAME    4K          4K     00000000-00000FFF
+-----+-----+-----+-----+-----+-----+

```

Figure 38. Typical output from VMAP command

In this example, the V=V region overlaps the V=R region.

Note that if the Read/Write value is not divisible by 1024, it appears in bytes.

Frame Information

There are two types of frame minor commands: one for areas of virtual storage, and one for RSM's internal queues.

An F in the argument field of the virtual storage area frame minor commands displays the number of fixed frames. For example, FTOT displays the total frames in the system; FTOTF displays the total *fixed* frames in the system. For the RSM internal queue frame minor commands, the F operand has no meaning.

The frame information minor commands of SYS display the real frames that back up virtual storage.

DFRS

Displays number of deferred frame allocations.

FAFQ

Displays available RSM internal queue frames.

FAVL

Displays available RSM internal queue frames.

FBDF

Displays Bottom Double (BDF) RSM internal queue frames.

Real storage frames used for segment table entries.

FBLD

Displays BLDL list frames.

FCOM

Displays common area frames.

This includes all CSA, SQA, LPA, and nucleus frames less than 16 MB.

FCSA

Displays Common Service Area (CSA) frames.

FDFD

Displays RSM internal deferred FREEMAIN queue frames.

FDRF

Displays the number of DREF frames in real storage.

FDRF displays data in K. You can add a plot of data by using the plot character c before the command.

FDSM

Displays storage being used by the system for data space management.

FDSM accepts a plot character in column 1.

FDSM displays the number of frames followed by the storage amount in kilobytes.

FDSP

Displays real storage in use by data spaces.

FDSP accepts a plot character in column 1.

This command displays the number of frames followed by the storage amount in kilobytes.

FECM

Displays extended common area frames.

This includes all CSA, SQA, LPA, etc. more than 16 MB.

FECS

Displays extended Common Service Area (CSA) frames.

FEFL

Displays extended Fixed Link Pack Area (FLPA) frames.

FELS

Displays extended Local System Queue Area (LSQA) frames.

FEML

Displays extended Modified Link Pack Area (MLPA) frames.

FEPL

Displays extended Pageable Link Pack Area (PLPA) frames.

FEPV

Displays extended private area frames.

FERO

Displays extended read-only nucleus frames.

FERW

Displays extended read/write nucleus frames.

FESQ

Displays extended SQA frames.

FFLP

Displays Fixed Link Pack Area (FLPA) frames.

FFLW

Displays RSM internal flawed frames.

FHSA

Displays hardware system area (HSA) frames.

FIDA

Displays RSM internal IDA frames.

FLSQ

Displays Local System Queue Area (LSQA) frames.

FMLP

Displays Modified Link Pack Area (MLPA) frames.

FOFL

Displays offline frames.

FPLP

Displays Pageable Link Pack Area (PLPA) frames.

FPVT

Displays private area frames.

FRON

Displays read-only nucleus frames.

FRWN

Displays read/write nucleus frames.

FSBF

Displays RSM internal SDUMP buffer frames.

FSQA

Displays SQA frames.

FSRS

Displays RSM internal SQA reserve queue frames.

FSYS

Displays system area frames.

FTDF

Displays RSM internal top double (TDF) queue frames.

FTOT

Displays total frames.

Real storage frames used for segment table entries.

FUNI

Displays RSM internal uninitialized frames.

FUNQ

Displays RSM internal unqueued frames.

FVIO

Displays RSM internal VIO frames.

FVR

Displays virtual=real (V=R) frames.

FVRW

Displays RSM internal frames waiting for V=R memory.

FVV

Displays Virtual=Virtual (V=V) frames.

Hiperspace commands**HSPI**

Displays the number of Hiperspace pages paged-in from auxiliary storage.

The data HSPI displays may be more meaningful if you use the .D (difference) or .R (rate) argument.

HSPO

Displays the number of Hiperspace pages paged-out to auxiliary storage.

The data HSPO displays may be more meaningful if you use the .D (difference) or .R (rate) argument.

HSPM

Displays the total number of Hiperspace pages paged-in from and paged-out to auxiliary storage.

HSPM = HSPI + HSPO.

The data HSPM displays may be more meaningful if you use the .D (difference) or .R (rate) argument.

Swap count commands

The SYS swap count minor commands display swap count information. Swap count numbers are stored in a wraparound field in z/OS, so the numbers might be meaningless. Therefore, it is better to show the value as a rate of pages/second: to do this, append .R to these commands.

A symbol in column 1 becomes the plot symbol, and the plot has a maximum of 50 points.

SCAL

Displays the number of physical swap-outs.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCDW

Displays the number of swap-outs due to detected waits.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCEX

Displays the number of physical swap-outs due to exchange swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCIR

Displays the number of swaps done to recover real storage frames from swappable address spaces.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCLF

Displays the number of attempted logical swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCLS

Displays the number of logical swap-outs.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCLW

Displays the number of swap-outs due to long waits.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCNQ

Displays the number of physical swap-outs due to enqueue swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCOI

Displays the number of swaps done due to OpenMVS (UNIX System Services) input waits.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCOO

Displays the number of swaps done due to OpenMVS (UNIX System Services) output waits.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCRQ

Displays the number of physical swap-outs due to request swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCRS

Displays the number of physical swap-outs due to RSM swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCTI

Displays the number of swap-outs due to terminal input swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCTO

Displays the number of swap-outs due to terminal output swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCTS

Displays the number of physical swap-outs due to TRANSWAP swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCUS

Displays the number of physical swap-outs due to unilateral swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SCXS

Displays the number of physical swap-outs due to ASM swaps.

This command accepts a plot symbol in column 1.

To express the value as a rate, append .R to the command.

SRM information

The SRM considers a number of factors in a system-wide look at resource utilization.

- ASM Queue length
- Percentage below 16M fixed
- CPU utilization thresholds for demand paging analysis

- CPU utilization
- Demand paging rate
- Page fault rate
- Paging delay time thresholds for demand paging analysis
- Paging delay time in milliseconds
- Total paging rate
- Unreferenced interval count
- Percentage of total real storage fixed

Some SRM information commands show how the current values affect SRM adjustments to the system. In [Figure 39 on page 138](#), this information appears in the far right columns.

```

SYS    >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
RCT    Field ----- --Value-- --Low-- --High-- --Effect--
CPGR   Common Page-in Rate          5      2      4 *WSS-INC*
CWSS   Common Target WKST           50     50    32767
RBEL   % Fixed Frames (below 16Meg)  28     82     88 *MPL-INC*
RCPU   CPU Utilization              55    101    101 *MPL-INC*

```

Figure 39. Typical output for SRM command

The possible values in the Effect column are:

blank

SRM does not use this value for decision-making, or high-low thresholds do not exist.

MPL-INC

Multiprogramming level is increased.

MPL-DEC

Multiprogramming level is decreased.

PROCE-INC

Number of processors enabled for I/O interruptions is increased.

PROCE-DEC

Number of processors enabled for I/O interruptions is decreased.

THNK-INC

Think time is increased.

THNK-DEC

Think time is decreased.

WSS-INC

Working set size is increased.

WSS-DEC

Working set size is decreased.

CPGR

Displays common area page-in rate.

Type:

Minor of SYS

This command shows how the current values affect SRM adjustments to the system.

CWSS

Displays common area target working set size in frames.

LUIC

Displays unreferenced interval count and threshold for logical swap think time adjustment.

This command displays current system-wide high UIC, as well as the happy values used to adjust the system think time, which the THINK minor of SYS displays.

RBEL

Displays percent of real storage less than 16 MB fixed.

This command shows how the current values affect SRM adjustments to the system.

RCPR

Displays average percentage CPU utilization as calculated by RMF.

This command displays the average percent CPU utilization computed by RMF. To use this command, RMF API support must be installed. In addition, the Monitor I CPU report must be active.

RCPU

Displays CPU utilization.

This command shows how the current values affect SRM adjustments to the system.

In an LPAR environment, SRM does not take into account LPAR non-dispatch time, which can lead to inflated values being reported. Use the RCPR command instead.

RCT

Provides title line.

RCT provides a title line under which the output for the rest of the commands described in this section can appear.

The Low and High values come from the user's PARMLIB. The Effect column states how the current values affect SRM adjustments to the system.

RCTA

Displays current SRM averages.

RCTA displays the Unreferenced Interval Count (UIC), page fault rate, CPU utilization, and ASMQ average values. The ASMQ is a queue of page I/O requests that are unserviced or incomplete.

The SRM averages these values over the last MPL adjustment interval. In an LPAR environment, SRM does not take into account LPAR non-dispatch time, which can lead to inflated values being reported. Use the RCPR command instead.

The average CPU utilization value calculated by the SRM varies from 0 to 100 percent, even if there are two or more processors active. The value is a utilization for the system as a whole. This value can actually be set by the SRM to 128 percent. This is a signal that, during the previous SRM interval, none of the processors ever entered the wait state, and more work (up to 28 tasks) is waiting for the CPU.

RCTH

Displays current SRM high thresholds.

UIC, paging, CPU utilization, and ASMQ high values appear.

The ASMQ is a queue of page I/O requests that are unserviced or incomplete.

The SRM averages this value over the last 20 seconds.

RCTI

Displays number of samples.

Usually thirty samples are taken.

RCTL

Displays current SRM low thresholds.

Unreferenced Interval Count (UIC), paging, CPU utilization, and ASMQ low values appear.

The ASMQ is a queue of page I/O requests that are unserviced or incomplete.

The SRM averages this value over the last 20 seconds.

RPAG

Displays page fault rate.

The page fault rate is the sum of page-ins and reclaims.

This command shows how the current values affect SRM adjustments to the system.

RREA

Displays percent of total real storage fixed.

This command shows how the current values affect SRM adjustments to the system.

RTPG

Displays total paging rate.

RTPI

Displays percent of I/O interrupts handled via TPI.

This command shows how the current values affect SRM adjustments to the system.

RUIC

Displays unreferenced interval count.

This command shows how the current values affect SRM adjustments to the system.

THNK

Displays system think time for logical swapping.

Miscellaneous system information minor commands

ALIBnn

Displays the current list of APF-authorized libraries.

An argument of *nn* suppresses the first *nn* lines of output.

The *ALI_{nnn}* command suppresses the first *nnn* lines of output.

ASVT

Displays status of Address Space Vector Table slots.

Status includes the following information:

- total number of slots
- number of in-use slots
- number of available slots
- number of unusable slots

If you add an argument of *U* to the ASVT command (*ASVTU*), the ASIDs of the unusable ASVT slots appear (eight per line).

The MAXU exception indicates when the number of active address spaces exceeds a threshold.

CPID

Displays CPU ID.

ENV

Displays system environmental information.

ENV displays the following information:

- The current level of z/OS
- The operating environment
- If applicable, the current LPAR or domain number
- The unit number of the processor
- The model number of the processor
- The serial number of the processor
- The date and time of the last IPL
- Whether RMF is active and its level
- The ESCON status of the processor

- Whether or not there is an ESCON director in the configuration
- Whether WLM GOAL mode is active.
- Whether HiperDispatch is active.
- Whether Series z Application Assist Processors (IFAs) are configured and how many.
- zAAP Crossover setting, if supported.
- zAAP Honor Dispatch Priority setting.
- zAAP processor speed relative to a standard CP.
- Whether System z® Integrated Information Processors (IIP) are configured and how many.
- zIIP Honor Dispatch Priority setting.
- zIIP processor speed relative to a standard CP.

ASID and FMID, if displayed, refer to the method by which OMEGAMON determined the RMF level.

```

----- ZLPAR1   VTM      OM/DEX   V420./C SYS 02/27/09 14:45:22
> Help PF1      Back PF3      Up PF7      Down

SYS  >> WLM Goal mode OPT=H0 SYSRES=(PRIPKT,7103) <<
env  z/OS  01.10.00 running in Logical partition 19 on 2097
+    IPLed at 10:46:00 on 02/24/09  RMF 7.1.1 is active  >> ASID <<
+    ESCON Status: Enabled  ESCON Director: In Configuration
+    WLM Mode=Goal          HiperDispatch Management: On
+    IFAs: 00                IFA Relative Processor Speed: 1.00:1
+                                IFA Honor Priority: Yes
+    IIPs: 02                IIP Relative Processor Speed: 1.00:1
+                                IIP Honor Priority: Yes

```

Figure 40. ENV command output with HiperDispatch enabled.

LLT

Displays link list libraries and their APF-authorization status.

Each line of output contains one data set name in the linklist concatenation.

LPAT

Displays list of libraries concatenated to SYS1.LPALIB.

Each line of output contains one data set name found in the linklist concatenation.

This command is designed to work on an unmodified IBM BLDL system; the results are unpredictable if you have another vendor product or usermod that alters the way BLDL is used.

LSFR

Displays number of real frames allocated to logically swapped address spaces.

LSNW

Displays number of logically swapped address spaces due to waits that are not terminal waits.

LSTW

Displays number of logically swapped address spaces due to terminal input or output waits.

PPT

Displays a list of programs in the Program Properties Table along with their special attributes.

See the IBM *MVS Initialization and Tuning Reference* for a description of the attributes.

RMSU

Displays 4-Hour Rolling Average Millions of Service Units per hour (MSU) information.

Format:

[X]RMSU[DB]

<blank>

Unmodified, displays summary 4-Hour Rolling Average MSU statistics.

X

Prefixed by an X, displays detailed 4-Hour Rolling Average MSU statistics.

DB

Suffixed by DB, displays hexadecimal and character dump of the RMSU data collection work area.

The values displayed by this command depend on platform support for the services that retrieve the data and whether License Manager is being used to define LPAR Capacity limits. RMSU displays the following information:

- The 4-Hour Rolling Average MSU.
- If the LPAR has a defined capacity set:
 - The percentage of time during the 4-Hour Rolling Average MSU period when the LPAR was capped by WLM.
 - The percentage of that time the LPAR was uncapped.
- The LPAR capacity in MSUs per Hour.
- The percentage of LPAR Capacity (or Defined LPAR Capacity) that the 4-Hour Rolling Average MSU value represents.
- If the LPAR is defined as a member of an LPAR Capacity Group:
 - The LPAR Group Name.
 - The LPAR Group Capacity in MSUs per Hour.
 - The percentage of the LPAR Group Capacity that the LPAR 4-Hour Rolling Average MSU value represents.
 - Whether or not the capacity limit is based on the LPAR's entitlement calculated as the number of standard CPs configured to the LPAR (including reserved CPs) multiplied by the MSUs/CP. If so a parenthesized "(Entitled)" is displayed. Alternatively, if a Defined Capacity is specified for the LPAR, a parenthesized literal "(Defined)" is displayed.
 - The average unused LPAR Group service units available over the last 4 hours, in millions of service units (MSUs).

This value is calculated using the service consumption of all members of the LPAR Group the current LPAR is a member of. If this value is negative, the LPAR Group members are currently subject to LPAR Group capping. The value is displayed in green if it represents more than 10% of the LPAR Group Capacity Limit, yellow if between 0.01 and 10%, and red at 0% or negative.
 - The current LPAR Capacity Limit enforced by the LPAR Group, in millions of service units per hour (MSUs).

This limit is affected by several factors:

- The value is only displayed if the LPAR is in an LPAR Group.
- If an LPAR Defined Capacity Limit is specified this value will never exceed that limit. If LPAR Group Capping is in effect, the value is calculated every 10 seconds and ranges from the LPAR Group entitlement minimum to a maximum of the LPAR Defined Capacity Limit.
- If no LPAR Defined Capacity is specified:
 - If the LPAR Group is not currently being capped, the value will be equal to the LPAR Group Capacity.
 - If LPAR Group Capping is in effect, the value is calculated every 10 seconds and ranges from the LPAR Group entitlement minimum to a maximum of the LPAR Group Capacity Limit.
 - If Group Capping is in effect, this value is displayed in red.


```

SYS    >> WLM Goal mode OPT=HD SYSRES=(PETPC0,3F22) <<
Xrmsu
+ LPAR 4-Hour Rolling Average MSU          304 (100.0% Capped, .0% Uncapped)
+ LPAR Capacity Limit (Entitled)          392 ( 77.6% LPAR Capacity utilized)
+
+ LPAR Group Name                          TESTPLEX
+ Group Capacity Limit (MSUs/Hour)        600 (Average Unused MSUs/Hour   -15)
+ Group LPAR MSU Limit (MSUs/Hour)        300
+
+
+      % Time Uncapped % LPAR % Time Capped % LPAR Unused
+ Time Period Uncapped MSUs/Hr Uncapped Capped MSUs/Hr Capped Group
+-----+-----+-----+-----+-----+-----+-----+-----
+ 18:14-18:14 .00 .00 .00 100.00 309.96 79.07 -19.87
+ 18:09-18:14 .00 .00 .00 100.00 306.73 78.25 -16.35
+ 18:04-18:09 .00 .00 .00 100.00 303.60 77.45 -13.67
+ 17:59-18:04 .00 .00 .00 100.00 298.21 76.07  -8.73
+ 17:54-17:59 .00 .00 .00 100.00 301.38 76.88 -11.46
+ 17:49-17:54 .00 .00 .00 100.00 300.47 76.65  -9.69
+ 17:44-17:49 .00 .00 .00 100.00 303.12 77.33 -12.77
+ 17:39-17:44 .00 .00 .00 100.00 302.59 77.19 -12.76
+ 17:34-17:39 .00 .00 .00 100.00 302.68 77.21 -12.54
+ 17:29-17:34 .00 .00 .00 100.00 306.05 78.07 -15.99
+ 17:24-17:29 .00 .00 .00 100.00 305.68 77.98 -15.66
+ 17:19-17:24 .00 .00 .00 100.00 304.31 77.63 -13.38
+ 17:14-17:19 .00 .00 .00 100.00 304.32 77.63 -14.30
+ 17:09-17:14 .00 .00 .00 100.00 307.73 78.50 -17.11
+ 17:04-17:09 .00 .00 .00 100.00 305.98 78.06 -16.24
+ 16:59-17:04 .00 .00 .00 100.00 305.46 77.92 -14.94
+ 16:54-16:59 .00 .00 .00 100.00 301.23 76.84 -11.38
+ 16:49-16:54 .00 .00 .00 100.00 305.17 77.85 -13.91

```

Figure 43. Typical output for XRMSU command if LPAR group capping is in effect

SMPP

Displays average time to transfer a swap page in milliseconds.

SQDC

Displays SQA deficit count.

SQRV

Displays number of times SQA reserve pool used.

SRB c

Displays contents of c.

c can be any of the following:

L

Contents of Local Service Manager Queue (LSMQ)

G

Contents of Global SMQ (GSMQ)

P

Contents of global Service Priority List (SPL)

SVCnnn,m

Displays address of SVC nnn; m is an optional route code.

Route codes are only required for router SVCs, and you must separate them with a comma from the SVC number.

To find the address for SVC 109 (a router SVC), enter SVC 109 followed by a route code, in this case, 7:

```
SVC 109,7
```

The result is:

```
SVC 109,7 (X'6D',X'07') is located at 00BD4000 Type: 3 or 4 APF-AUTH
```

SVCXxx,m

Displays address of SVC xx (hexadecimal); *m* is an optional route code.

Route codes are only required for router SVCs, and you must separate them with a comma from the SVC number.

To find the address for SVC 6D (a router SVC), enter SVCX6D followed by a route code, in this case, 7:

```
SVCX6D,7
```

The result is:

```
SVC 109,7 (X'6D',X'07') is located at 00BD4000 Type: 3 or 4 APF-AUTH
```

CPU utilization commands

MCPU nn

Displays CPU utilization greater than $nn\%$ by address space and system-wide.

Type:

Immediate

Format:

MCPU nn

The MCPU command displays utilization of standard (CP), Series z Application Assist (IFA), and System z Integrated Information (zIIP) processors by address space and individual processor. LPAR-wide TCB, SRB, IFA, IFA on CP, zIIP, zIIP on CP, and enclave utilization is also displayed.

Task

Shows CPU utilization by task or address space. Each address space's CPU usage over the last OMEGAMON interval is calculated in TCB and SRB time. The total of these two times is calculated as a percentage of the total time available (for one CPU). The maximum possible value for CPU utilization is $100\% \times$ number of CPUs being monitored. MCPU nn displays and plots any task whose CPU usage exceeds the $nn\%$ threshold. (If you *only* want to see Task plotted across the entire screen, use the SCPU nn command.)

System

Shows the total CPU utilization system-wide. The totals in this zone are calculated over the last OMEGAMON interval, and displayed in terms of TCB, SRB, IFA, IFA on CP (IFC), zIIP, zIIP on CP (IIC), and enclave utilization (NCL) time. The difference between the total software times and the total hardware time is attributed to z/OS overhead. In other words, these are non-billable hardware cycles because they are not accounted for in terms of SRB, TCB, zAAP, and zIIP times. On a PR/SM system, the PARx line are given to show average utilization logically for the partition.

The address spaces displayed can be filtered by the total processor utilization percentage. For example, the command XMCPU05 displays all address spaces where the CPU percentage across standard, IFA and zIIP utilization is greater than 5%. Note that IFA and zIIP time consumed by enclaves owned by the address space are *not* included in the **IFA%** and **IIP%** column data; the time is accumulated in the total enclave CPU percentage displayed after the NCL : literal. The system-wide IFA and zIIP values displayed after the IFA: and IIP: literals on the right of the output do not include owned enclave consumption. (The Address Space CPU Utilization displays in OMEGAMON for z/OS workspaces are more comprehensive, showing the owned enclave IFA and zIIP consumption for enclaves and distinguishing dependent and independent enclave consumption.)

MCPU shows CPU usage by task as well as overall CPU utilization. To view CPU resource consumption within an address space, place the cursor under a job name (in the **TASK** column) and press PF11. If the job name begins with an asterisk, then place the cursor on the character following the asterisk and press PF11. If your system is logically partitioned (for example, z990 with PR/SM support), place the cursor under the partition and press PF11 for more information on the partition.

To view more detailed enclave statistics by owning address space, place the cursor under the NCL : literal and press PF11. If your system supports zAAP processors, place the cursor under the IFA: or IFC: literal and press PF11 for more information on zAAP and zAAP on CP resource consumption. If your system supports zIIP processors, place the cursor under the IIP: or IIC: literal and press PF11 for more information on zIIP and zIIP On CP resource consumption.

The following figure displays output for the MCPUCPU command with APF-authorization.

```

MCPUCPU01|__Task__ CPU% 0_____ 100|__PG N/A in goal mode__|System % 0_____ 100|
+ |XCFAS 2.1 > . . . . .|TCB: 96 ----->>>|.
+ |CTG720A 1.3 > . . . . .|SRB: 3 > . . . . .|
+ |S3SUBK39 1.0 > . . . . .|NCL: 0 > . . . . .|
+ |PLGWCP7 79.2 ----->. |MVS: 12 ->. . . . .|
+ | | | |IFA: 0 > . . . . .|
+ | | | |IFC: 0 > . . . . .|
+ | | | |IIP: 0 > . . . . .|
+ | | | |IIC: 0 > . . . . .|
+ | | | |0-----100|
+ | | | |CPU00 41 ---=> . . . .|
+ | | | |CPU01 0 > . . . . .|
+ | | | |CPU02 70 ----->. |
+ | | | |IIP03 0 > . . . . .|
+ | | | |PAR17 37 --->. . . . .|

```

Figure 44. Typical output display for MCPUCPU with APF-authorization

OMEGAMON alters the high end of the scale to the first value large enough to depict the address space, or system with the largest CPU utilization percentage.

Prefixed with an X (XMCPUCPU), the command displays all tasks that use more than nn% of the CPU. (5% is the default. You can change it by typing over 05.) The center section of the display shows a breakdown of the CPU percentage into TCB, SRB, zAAP (IFA), zAAP on CP (IFC), zIIP (IIP) and zIIP on CP (IIC) consumption. The final section of the display breaks down CPU utilization into seven components:

- TCB: Task program execution
- SRB: System services on behalf of tasks
- NCL: Enclaves (including standard CP, zAAP and zIIP utilization)
- MVS: MVS overhead
- IFA: zAAP processor utilization (normalized)
- IFC: zAAP-eligible work executed on standard CPs (crossover)
- IIP: zIIP processor utilization (normalized)
- IIC: zIIP-eligible work executed on standard CPs (crossover)

```

XMCPUCPU01|----- ZMENU VTM OM/DEX V420./C SYS 03/10/09 15:40:15 1
+ |__Task__ CPU% TCB% SRB% IFA% IFC% IIP% IIC%|System % 0_____ 100|
+ |XCFAS 2.1 1.6 .5 .0 .0 .0 .0|TCB: 96 ----->>>|.
+ |CTG720A 1.3 1.3 .0 .0 .0 .0|SRB: 3 > . . . . .|
+ |S3SUBK39 1.0 .9 .1 .0 .0 .0|NCL: 0 > . . . . .|
+ |PLGWCP7 79.2 79.2 .0 .0 .0 .0|MVS: 12 ->. . . . .|
+ | | | |IFA: 0 > . . . . .|
+ | | | |IFC: 0 > . . . . .|
+ | | | |IIP: 0 > . . . . .|
+ | | | |IIC: 0 > . . . . .|
+ | | | |0-----100|
+ | | | |CPU00 41 ---=> . . . .|
+ | | | |CPU01 0 > . . . . .|
+ | | | |CPU02 70 ----->. |
+ | | | |IIP03 0 > . . . . .|
+ | | | |PAR17 37 --->. . . . .|

```


Note: OMEGAMON must be APF authorized for the CPU percentage to appear.

NCLV

Collects performance data for all completed and active enclaves on the system.

Type:

Major

Other than displaying the measurement interval that the collected data represents, the NCLV command does not display any data itself. It is used with the following minor commands:

EDPP

Displays CPU percentages for dependent enclaves.

EDPT

Displays CPU times for dependent enclaves.

EIDP

Displays CPU percentages for independent enclaves.

EIDT

Displays CPU times for independent enclaves.

ETTP

Displays CPU percentages for combined independent and dependent enclaves.

ETTT

Displays CPU times for combined independent and dependent enclaves.

INT

Allows a measurement interval, in minutes, to be specified. An interval can be between 1 and 998 minutes, open-ended, or from "Enter" to "Enter".

EDPP

Displays dependent enclave information as percentages by owning address space.

Type:

Minor of NCLV

Format:

[U]EDPP [xxxx]

<blank>

Unmodified, EDPP displays normalized statistical values.

U

Prefixed by a U, EDPP displays unnormalized statistical values.

nnnn

Followed by nnnn, EDPP displays statistical values where address space owned enclave CPU consumption is greater than or equal to nnnn%..

EDPP displays the following information for each enclave-owning address space where its dependent or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource percentage consumed by enclaves.
- zAAP resource percentage consumed by enclaves.
- zIIP resource percentage consumed by enclaves.
- Standard CP resource percentage consumed by enclaves performing zAAP-eligible work.
- Standard CP resource percentage consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP percentage consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

EDPT

Displays dependent enclave information as times by owning address space.

Type:

Minor of NCLV

Format:

[U]EDPT [nnnn]

<blank>

Unmodified, EDPT displays normalized statistical values.

U

Prefixed by a U, EDPT displays unnormalized statistical values.

nnnn

Followed by nnnn, EDPT displays statistical values where address space owned enclave CPU consumption is greater than or equal to *nnnn*%.

EDPT displays the following information for each enclave-owning address space where its dependent or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource time consumed by enclaves.
- zAAP resource time consumed by enclaves.
- zIIP resource time consumed by enclaves.
- Standard CP resource time consumed by enclaves performing zAAP-eligible work.
- Standard CP resource time consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP time consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

EIDP

Displays CPU percentages for independent enclaves.

Type:

Minor of NCLV

Format:

[U]EIDP [nnnn]

<blank>

Unmodified, EIDP displays normalized statistical values.

U

Prefixed by a U, EIDP displays unnormalized statistical values.

nnnn

Followed by nnnn, EIDP displays statistical values where address space owned enclave CPU consumption is greater than or equal to *nnnn*%.

EIDP displays the following information for each enclave-owning address space where its dependent and/or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource percentage consumed by enclaves.
- zAAP resource percentage consumed by enclaves.
- zIIP resource percentage consumed by enclaves.
- Standard CP resource percentage consumed by enclaves performing zAAP-eligible work.
- Standard CP resource percentage consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP percentage consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

EIDT

Displays independent enclave information as times by owning address space.

Type:

Minor of NCLV

Format:

[U]EIDT [nnnn]

<blank>

Unmodified, EIDT displays normalized statistical values.

U

Prefixed by a U, EIDT displays unnormalized statistical values.

nnnn

Followed by nnnn, EIDT displays statistical values where address space owned enclave CPU consumption is greater than or equal to *nnnn*%.

EIDT displays the following information for each enclave-owning address space where its dependent or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource time consumed by enclaves.
- zAAP resource time consumed by enclaves.
- zIIP resource time consumed by enclaves.
- Standard CP resource time consumed by enclaves performing zAAP-eligible work.
- Standard CP resource time consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP time consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRIT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

ETTP

Displays CPU percentages for combined independent and dependent enclaves by owning address space.

Type:

Minor of NCLV

Format:

[U]ETTP [nnnn]

<blank>

Unmodified, ETTP displays normalized statistical values.

U

Prefixed by a U, ETTP displays unnormalized statistical values.

nnnn

Followed by nnnn, ETTP displays statistical values where address space owned enclave CPU consumption is greater than or equal to *nnnn*%.

ETTP displays the following information for each enclave-owning address space where its dependent and/or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource percentage consumed by enclaves.
- zAAP resource percentage consumed by enclaves.
- zIIP resource percentage consumed by enclaves.
- Standard CP resource percentage consumed by enclaves performing zAAP-eligible work.
- Standard CP resource percentage consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP percentage consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

ETTT

Displays total enclave information as times by owning address space.

Type:

Minor of NCLV

Format:

[U]ETTT [nnnn]

<blank>

Unmodified, ETTT displays normalized statistical values.

U

Prefixed by a U, ETTT displays unnormalized statistical values.

nnnn

Followed by nnnn, ETTT displays statistical values where address space owned enclave CPU consumption is greater than or equal to *nnnn*%.

ETTT displays the following information for each enclave-owning address space where its dependent and/or independent enclaves have consumed CPU resource over the measurement interval:

- Address space name.
- Address space ID in hexadecimal format.
- Standard CP resource percentage consumed by enclaves.
- zAAP resource percentage consumed by enclaves.
- zIIP resource percentage consumed by enclaves.
- Standard CP resource percentage consumed by enclaves performing zAAP-eligible work.
- Standard CP resource percentage consumed by enclaves performing zIIP-eligible work.
- Total of combined standard CP, zAAP and zIIP percentage consumed by enclaves.

A totals line displaying the sum of these values by column is displayed after the individual address space statistics.

Statistical values can be thresholded as follows:

- Address Space Level Warning set by POPT Major Command value SCPUWARN.
- Address Space Level Critical set by POPT Major Command value SCPUCRIT.
- LPAR Level Warning set by POPT Major Command value SPCUTWARN.
- LPAR Level Critical set by POPT Major Command value SPCUTCRT.

Exceeding Warning thresholds results in a statistical value being highlighted in the color Yellow. Exceeding Critical thresholds results in a statistical value being highlighted in the color Red. Note that statistics where there has been activity over the measurement interval are highlighted in the color green.

INT

Sets measurement interval for NCLV major command statistical minors.

Type:

Minor of NCLV

Format:

INT [nnn]

Used without operands, this command displays the measurement interval currently in effect. Valid values for *nnn* are 0-999, inclusive. Values 1 through 998 represent the number of minutes before a new measurement interval will start. A value of 0 sets the interval as "Enter" to "Enter". A value of 999 sets an indefinite interval. Resetting the measurement interval will result in the NCLV command data collection re-initializing.

SCPU nn

Displays address spaces using more than *nn*% CPU since the last OMEGAMON cycle.

Type:

Immediate

For each address space selected, SCPU nn displays:

- CPU utilization (including but not limited to TCB, SRB, and other CPU measurements such as enclave, Series z Application Assist Processors, and System z Integrated Information Processors time) over the last cycle.
- A plot of CPU utilization. OMEGAMON alters the high end of the scale to the first value large enough to depict the address space with the largest CPU utilization.

The last line of the display is total CPU utilization, not including z/OS overhead, calculated from the sum of all address space utilizations over the last OMEGAMON cycle. The selection threshold (*nn*) does not affect this total.

This total is probably not the same as the value that the RCTA minor command reports, because RCTA calculates CPU utilization from machine wait time, averaged over an SRM-defined interval of up to 20 seconds, depending upon CPU models.

```

SCPU02  __CPU Utilization__  0__20__40__60__80__100__120__140__160__180__200
+   *MASTER*      7.1  |-> . . . . . . . . . . . . . . . .
+   KOBVTAMA      5.6  |-> . . . . . . . . . . . . . . . .
+   TDOM29        2.7  |> . . . . . . . . . . . . . . . .
+   TDOC07B       3.5  |> . . . . . . . . . . . . . . . .
+   CANDLIMS     112.1 |---20---40---60===80==100=> . . . . . .
+   CICS          56.2 |---20---40--> . . . . . . . . . . . .
+   Total        206.4 |---20---40---60===80==100==120==140++160++180++200>
In this display, nn = 02, therefore all active address spaces using more
than 2% CPU appear.

```

Figure 45. Typical output for SCPU command

SIFAnn

Displays IFA and IFA on CP utilization of address spaces using more than *nn*% IFA and IFA on CP combined.

Type:

Immediate

Format:

SIFAnn

where *nn* specifies the threshold over which percentages of IFA and IFA on CP are displayed.

The SIFA command displays utilization percentages of Series z Application Assist Processors (zAAPs: for reporting purposes, referred to as an integrated facility for applications, or IFA%) and utilization of standard processors by zAAP-eligible work (IFA% on CP) for each address space and a total value for all enclaves.

For each address space with an IFA and IFA on CP combined percentage over the threshold, SIFA displays the IFA utilizations since the last cycle and a utilization plot. If there is active enclaves IFA

work, an enclaves entry provides the total IFA utilization and IFA on CP utilization of all dispatcher enclaves on the z/OS image.

The Total entry at the end of the output is the sum of IFA and IFA on CP percentages for all address spaces and enclaves on the z/OS image. The threshold selected (*nn*) does not affect this total.

SIIPnn

Displays zIIP and zIIP on CP utilization of address spaces using more than *nn*% zIIP and zIIP on CP combined.

Type:

Immediate

Format:

```
SIIPnn
```

where *nn* specifies the threshold over which percentages of zIIP and zIIP on CP combined are displayed.

SIIP displays utilization of System z Integrated Information Processors (zIIP%) and utilization of standard processors by zIIP-eligible work (zIIP% on CP) for each address space and a total value for all enclaves. For each address space with a zIIP and zIIP on CP combined percentage over the threshold, SIIP displays the zIIP utilizations since the last cycle and a utilization plot. If there is active enclaves zIIP work, an Enclaves entry provides the total zIIP utilization and zIIP on CP utilization of all dispatcher enclaves on the z/OS image.

The Total entry at the end of the output is the sum of zIIP and zIIP on CP percentages for all address spaces and Enclaves on the z/OS image. The threshold selected (*nn*) does not affect this total.

System information commands

The HDSP and LPAR immediate commands provide information about logical process utilization and status for HiperDispatch and about logical partitions and LPAR management overhead.

HDSP

Displays details of logical processor utilization and status for HiperDispatch.

Type:

Immediate

HDSP displays HiperDispatch statistics and status. Information provided by the HDSP command is as follows:

- Header Information:
 - HiperDispatch Status
 - LPAR Name
 - LPAR Cluster Name or N/A
 - LPAR Group Name, if the LPAR is a member of an LPAR Group, or N/A

If LPAR Weight Management is enabled, the following information is displayed:

- Current, minimum, and maximum LPAR Weight for standard CP, Series z Application Assist Processors (zAAPs) and System z Integrated Information Processor (zIIPs)
- The measurement interval the statistical data represents, in seconds to 3 decimal places

If HiperDispatch management is active, the following information is displayed for all processor types across the z/OS image:

- System MVS%
- System PCPD%

Processor-level statistics are grouped by processor type: standard CP, zAAP, or zIIP. Each group is preceded by summary statistics:

- The number of processors of the specific type that are available
- The number of High HiperDispatch Priority processors
- The number of Medium HiperDispatch Priority processors
- The number of Low HiperDispatch Priority processors
- The number of processors in Parked status

The following information is displayed for each processor:

- Logical CPU ID as a 2-character hexadecimal field.
- HiperDispatch Priority as High, Medium, or Low.
- HiperDispatch Share Percentage. This value will be 100.0 for HiperDispatch Priority High, 0.0 for Low and any value in the range between these two values for Medium.
- MVS Percentage
- Parked Percentage
- Physical CPU utilization as a percentage to 3 decimal places.
- Management overhead as a percentage to 3 decimal places if LPAR management data is available.
- Processor Status as Parked, Park Pending, Online, Offline, or Reserved.


```

----- ZHDSP   VTM       OM/DEX   V420./C Z2   09/21/09 17:53:30
> Help PF1      Back PF3      Up PF7       Down PF8     Zoom PF11
-----
>
      HIPERDISPATCH STATISTICS AND STATUS

> HDSP shows HiperDispatch statistics and status for the current LPAR.

HDSP
+HiperDispatch Management: On
+LPAR Name: Z2          LPAR Cluster: ZPETPLX2      LPAR Group: N/A
+LPAR Weights (Curr./ Min./ Max.)      Measurement Interval: 1.006
+Standard CPs:      60/   40/   80
+   zAAP:           60/   40/   80
+   zIIP:           75/   75/   75
+
+System MVS% :      Standard CP:   8.180  zAAP:   3.780  zIIP:   .000
+System PCPD%:     Standard CP:   1.571  zAAP:   .646  zIIP:   .000
+
+ Standard CPs (Available= 10, High=  2, Medium=  1, Low=  7, Parked=  7)
+ LCPU HiperDsp
+ ID  Priority  Share %   MVS %   Park %   PCPD %   Mgmt %   Status
+ ---  -
+ 00  High    100.0    5.070   .000    4.871   .497    Online
+ 01  High    100.0    7.256   .000    7.058   .398    Online
+ 02  Medium  92.1     4.274   .000    3.777   .199    Online
+ 03  Low     .0        .000   100.000 .000    .000    Parked
+ 04  Low     .0        .000   100.000 .000    .000    Parked
+ 05  Low     .0        .000   100.000 .000    .000    Parked
+ 06  Low     .0        .000   100.000 .000    .000    Parked
+ 07  Low     .0        .000   100.000 .000    .000    Parked
+ 08  Low     .0        .000   100.000 .000    .000    Parked
+ 09  Low     .0        .000   100.000 .000    .000    Parked
+
+ zAAPs          (Available=  6, High=  0, Medium=  1, Low=  3, Parked=  2)
+ LCPU HiperDsp
+ ID  Priority  Share %   MVS %   Park %   PCPD %   Mgmt %   Status
+ ---  -
+ 0A  Medium  32.0     .000    .000    2.187   .099    Online
+ 0B  Low     .0        .000   100.000 .000    .000    Parked
+ 0C  Low     .0        .000   100.000 .000    .000    Parked
+ 0D  Low     .0        7.058   .000    .398    .000    Online
+ 10  Low     .0
+ 11  Low     .0
+
+ zIIPs          (Available=  2, High=  0, Medium=  1, Low=  1, Parked=  0)
+ LCPU HiperDsp
+ ID  Priority  Share %   MVS %   Park %   PCPD %   Mgmt %   Status
+ ---  -
+ 0E  Medium  16.2     .000   100.000 .000    .000    Online
+ 0F  Low     .0        .000   100.000 .000    .000    Online

```

Figure 46. Typical output for HDSP command

LPARnn

Displays data about logical partitions and LPAR management overhead.

Type:

Immediate

Format:

LPARnn

where *nn* is the number of the logical partition (1-7). If a logical partition (*nn*) is not specified for LPAR, data for all configured logical partitions is displayed.

APF-authorization is required.

LPAR displays current information about one or more PR/SM logical partitions. The information displayed includes:

- General information, including:
 - Model Number (Model).
 - The CPC model type-number (or type-version-code for old models).

On later model processors this is the Model Capacity ID which represents the current capacity delivered to the CPC, not the maximum capacity that can be brought into service, using reserved CPs, by OOCoD, CUoD, CPM or CBU etc.

- Serial Number (Serial). The CPC Serial number.
- CEC MSUs capacity. This is the CEC/CPC capacity in millions of service units per hour. This capacity represents the current capacity that can be delivered to the CPC, not including capacity that may be brought into service, using reserved CPs, by OOCoD, CUoD, CPM etc.
- Number of physical processors.
- Dispatch interval.
- Number of physical processors by type (CP or special CP: .ICFs, IFLs ,zAAPs or zIIPs).

If special processors exist, their logical partitions are displayed separately and following the CP section. It is possible that a partition displays in both sections. On IBM z9[®] and later processors, statistics for each special processor type are presented as a separate pool. Logical partitions are displayed in each pool where that pool's processor type is configured to that logical partition.

- Number of configured partitions, if you specify a single partition.
- Elapsed interval.

This is the wall-clock time since the last screen refresh. Utilization statistics are for this period.

- Name and number of the LPAR. An asterisk (*_ immediately to the left of the LPAR number highlights the LPAR on which OMEGAMON is running.
 - LPAR Status: Active or Deactive
- If supported by the hardware, statistics relevant to On/Off Capacity on Demand (OOCoD), Capacity Provisioning Management (CPM) and Capacity BackUp (CBU) may be displayed. These statistics are:

- Model Permanent Capacity ID.

The content of the this identifier corresponds to the capacity of the CPC exclusive of temporarily available increased capacity (OOCoD or CPM) and temporarily available replacement capacity (CBU).

- Model Permanent Capacity Rating. The capacity, in millions of service units (MSUs), of the type-model of the processor represented by the Model Permanent Capacity ID.
- Model Temporary Capacity ID.

The content of this identifier corresponds to the total of permanent capacity and temporarily available increased capacity (OOCoD or CPM) but exclusive of any temporarily available replacement capacity (CBU).

- Model Temporary Capacity Rating.

The capacity, in MSUs, of the type-model of the processor represented by the Model Temporary Capacity ID.

- OOCoD and CPM Adjustments.

This is an indicator (Yes or No) of whether or not CP resources have been added by OOCoD and CPM in the current configuration. The setting of the indicator is inferred from the fact that the Model Capacity ID and the Model Temporary Capacity ID are equal to each other, but greater than the Model Permanent Capacity ID. The indicator is also set to Yes if all three Capacity IDs have different values because this means that CBU is also in effect.

- CBU Adjustments.

This is an indicator (Yes or No) of whether or not CP resources have been added, or subtracted, by CBU in the current configuration. The setting of the indicator is inferred from the fact that the Model Temporary Capacity ID and Model Permanent Capacity ID are equal to each other but the Model Capacity ID is higher or lower than the two equal values. The indicator is also set to Yes if all 3 Capacity IDs have different values because this means that OOCoD and CPM is also in effect.

Information provided by LPAR (without an operand) includes the following:

- Weight Factor (Wgt). In shared physical processor environments, this indicates the relative importance of the LPAR regarding access to CPU cycles. MIX in this field indicates that processors in this LPAR have different weight factors
- Wait Completion indicator (WAIT) defined for the LPAR (Yes/No)
- CPU capping (CAP) indicator (Yes, No or xx%); xx% is the defined (soft) capacity in percent form
- Logical Processors (#LP). Average number of logical processors online
- Logical Processor dispatch time (LCPD%). Percentage of time dispatched for online logical processors
- Effective Physical Processor dispatch time (PCPD%). Percentage of time the LPAR was dispatched on a physical processor, excluding LPAR management time
- Physical Processor Overhead (OVHD%). Percentage of time utilized for LPAR management.
- Dispatch time. Sum of Effective Physical plus LPAR Management processor time utilized by the partition. The line containing logical partition named PHYSICAL displays only LPAR management overhead that is not attributable to a particular partition.

```

LPAR
+
+ Model = 2084-D32 Serial = 01B36A CEC MSUs capacity = 1188
+ Number of Physical processors = 32 Dispatch interval = DYNAMIC
+ Number of CPs = 27 Special CPs = 5 Elapsed interval = 00.00.01.299
+
+Name Number Stat Wgt Wait Cap #LP LCPD% PCPD% OVHD% Dispatch time
+CANSP22 *28 A 002 NO NO 3.0 31.17 3.46 .02 00.00.01.223
+RALNS1 D
+RALNS17 14 A 003 NO NO 2.0 7.54 .56 .01 00.00.00.200
+RALNS18 08 A 003 NO NO 2.0 80.83 5.99 .01 00.00.02.102
+RALNS19 09 A 003 NO NO 2.0 5.23 .39 .01 00.00.00.139
+RALNS2 D
+RALNS21 03 A 010 NO NO 1.0 .08 .00 .00 00.00.00.001
+RALNS33 10 A 003 NO NO 4.0 15.03 2.23 .02 00.00.00.787
+RALNS34 04 A 003 NO NO 4.0 22.03 3.26 .03 00.00.01.157
+RALNS35 05 A 003 NO NO 4.0 .63 .09 .01 00.00.00.037
+RALNS36 06 A 003 NO NO 4.0 .54 .08 .01 00.00.00.032
+RALVMQ 07 A 015 NO NO 12.0 20.36 9.05 1.02 00.00.03.531
+TIVVM1 12 A 018 NO NO 8.0 15.88 4.71 .34 00.00.01.771
+TIVVM4 11 A 018 NO NO 8.0 47.64 14.12 .23 00.00.05.031
+TIVVM7 15 A 010 NO NO 4.0 27.46 4.07 .04 00.00.01.442
+CANSP11 27 A 001 NO NO 2.0 2.81 .21 .01 00.00.00.078
+CANSP12 22 A 002 NO NO 3.0 3.67 .41 .02 00.00.00.150
+CANSP13 29 A 002 NO NO 1.0 17.78 .66 .02 00.00.00.239
+CANSP23 20 A 001 NO NO 2.0 2.81 .21 .01 00.00.00.078
+CANSYSL 30 A 002 NO NO 2.0 27.87 2.06 .02 00.00.00.730
+RALHCD D
+TIVMVS18 D
+PHYSICAL 00 4.08 00.00.01.432
+
+-----
+TOTAL CP 102 51.56 5.91 00.00.20.160
+
+
+RALVMQ 07 A .0 .00 .00 .00 00.00.00.000
+CANSP12 22 A 002 NO NO 1.0 .00 .00 .00 00.00.00.000
+CANSYSL 30 A 002 NO NO 1.0 .00 .00 .00 00.00.00.000
+PHYSICAL 00 1.15 00.00.00.075
+
+-----
+TOTAL ICF 4 .00 1.15 00.00.00.075

```

Figure 47. Typical output for LPAR command

An X in the label column (1) produces a modified LPAR report containing extended LPAR information related to the Intelligent Resource Director (IRD):

- Current (physical) LPAR weight percentage (Wgt%)
- Logical Weight Percent (LWgt%), calculated as:

$$\text{LWgt\%} = \text{Physical Weight \%} \times \# \text{ Physical Processors} / \# \text{ Logical Processors}$$

- Name of the cluster that this LPAR is assigned to, also known as the Sysplex name (CLUSTERNm)
- Demand Percent (Demd%). Ratio of effective physical utilization to target utilization, calculated as:

$$\text{Dmnd\%} = (\text{Physical CPU\%} * 100) / \text{Current Weight\%} - \text{Velocity (Vel\%)}$$

- Effective logical processor speed, also known as Effective Weight Percent); calculated as:

$$\text{Vel\%} = (\text{LCPD\%} * 100) / (\text{LCPD\%} + \text{Ready\%}) - \text{Ready\%}$$

- Percent of time when LPAR had "ready" work and was not dispatched
- Percent of time when LPAR was idle, that is, had no work to do (PSW Wait bit was on)

```

XLPAR
+      Model = 2084-D32  Serial = 01B36A      CEC MSUs capacity = 1188
+      Number of Physical processors = 32      Dispatch interval = DYNAMIC
+      Number of CPs = 27 Special CPs = 5      Elapsed interval = 00.00.01.299
+
+Name   Number  St  Wgt%  LWgt%  ClusterNm  #LP  Demd%  Vel%  LCPD%  Ready%  Wait%
+CANSP22 *28    A   2.0   18.0  LPAR400J   3.0  173.0  68.9  31.17  14.08  54.75
+RALNS1  D
+RALNS17  14    A   2.9   39.2  NIVTPLEX   2.0   19.3   29.3   7.54  18.21  74.25
+RALNS18  08    A   2.9   39.2  NIVTPLEX   2.0  206.6   91.8  80.83   7.24  11.93
+RALNS19  09    A   2.9   39.2  NIVTPLEX   2.0   13.4   46.2   5.23   6.08  88.68
+RALNS2  D
+RALNS21  03    A   9.8  264.6  N/A         1.0    .0  100.0   .08   .00  99.92
+RALNS33  10    A   2.9   19.6  PORPLX     4.0   76.9   52.0  15.03  13.85  71.12
+RALNS34  04    A   2.9   19.6  PORPLX     4.0  112.4   34.8  22.03  41.29  36.68
+RALNS35  05    A   2.9   19.6  PORPLX     4.0   3.1    5.7   .63  10.39  88.97
+RALNS36  06    A   2.9   19.6  PORPLX     4.0   2.8    3.2   .54  16.57  82.89
+RALVMQ   07    A  14.7   33.1  N/A        12.0   61.6   21.6  20.36  73.99   5.65
+TIVVM1   12    A  17.6   59.4  N/A         8.0   26.8   17.6  15.88  74.33   9.78
+TIVVM4   11    A  17.6   59.4  N/A         8.0   80.2   52.6  47.64  42.94   9.42
+TIVVM7   15    A   9.8   66.2  N/A         4.0   41.5   38.6  27.46  43.66  28.88
+CANSP11  27    A   1.0   13.5  LPAR400J   2.0   21.0   22.2   2.81   9.82  87.37
+CANSP12  22    A   2.0   18.0  LPAR400J   3.0   20.5   18.1   3.67  16.65  79.68
+CANSP13  29    A   2.0   54.0  LPAR400J   1.0   33.0   46.9  17.78  20.17  62.05
+CANSP23  20    A   1.0   13.5  LPAR400J   2.0   21.0   36.9   2.81   4.81  92.38
+CANSYSL  30    A   2.0   27.0  LPAR400J   2.0  103.0   64.8  27.87  15.13  57.01
+RALHCD   D
+TIVMVS18 D
+PHYSICAL 00
+
+      -----
+TOTAL CP          100.0
+
+
+RALVMQ   07    A           N/A         .0
+CANSP12  22    A  50.0  250.0  LPAR400J   1.0    .0    .0   .00   .00100.00
+CANSYSL  30    A  50.0  250.0  LPAR400J   1.0    .0    .0   .00   .00100.00
+PHYSICAL 00
+
+      -----
+TOTAL ICF          100.0

```

Figure 48. Typical output for XLPAR command

A Y in the label column (1) produces a modified LPAR report containing extended LPAR information related to the Workload Manager (WLM) CPU Management capability:

- Whether the LPAR is enabled for IRD WLM weight management (Yes or No)
- Current Weight (CurW)
- Weight at time of LPAR initialization (IniW)
- Maximum defined weight for the LPAR (MaxW)
- Minimum defined weight for the LPAR (MinW)
- Number of logical processors online for the LPAR (OnLP)
- Number of logical processors offline for the LPAR (OffP)

- Number of logical processors that are defined to the LPAR and could be varied online via operator intervention (SbyP)
- Number of logical processors that are defined to the LPAR but cannot varied online (svP). A Capacity Upgrade on Demand (CUoD) could convert one or more Reserved to Standby logical processors.
- Number of logical processors that have not been included in the LPAR definition (UndP)
- Amount of real storage (in megabytes) defined to the LPAR (Stor)

```

YLPAR
+      Model = 2084-D32  Serial = 01B36A      CEC MSUs capacity = 1188
+      Number of Physical processors = 32      Dispatch interval = DYNAMIC
+      Number of CPs = 27 Special CPs = 5      Elapsed interval = 00.00.01.299
+
+Name      Number  St  WgtM  CurW  InitW  MinW  MaxW  OnLP  OffP  SbyP  RsvP  UndP  Stor
+CANSP22   *28    A  NO    002    002    002    002    3.0   .0    1.0   .0  27.0  1536M
+RALNS1                    D                                N/A
+RALNS17   14    A  NO    003    003    003    003    2.0   .0    .0    .0  29.0  4096M
+RALNS18   08    A  NO    003    003    003    003    2.0   .0    .0    .0  29.0  4096M
+RALNS19   09    A  NO    003    003    003    003    2.0   .0    .0    .0  29.0  4096M
+RALNS2                    D                                N/A
+RALNS21   03    A  NO    010    010    010    010    1.0   .0    1.0   .0  29.0  8192M
+RALNS33   10    A  NO    003    003    003    003    4.0   .0    .0    .0  27.0  10240M
+RALNS34   04    A  NO    003    003    003    003    4.0   .0    .0    .0  27.0  6144M
+RALNS35   05    A  NO    003    003    003    003    4.0   .0    .0    .0  27.0  10240M
+RALNS36   06    A  NO    003    003    003    003    4.0   .0    .0    .0  27.0  6144M
+RALVMQ    07    A  NO    015    015    015    015   12.0   .0    4.0   .0  15.0  73728M
+TIVVM1    12    A  NO    018    018    018    018    8.0   .0    4.0   .0  19.0  36864M
+TIVVM4    11    A  NO    018    018    018    018    8.0   .0    4.0   .0  19.0  51200M
+TIVVM7    15    A  NO    010    010    010    010    4.0   .0    4.0   .0  23.0  12288M
+CANSP11   27    A  NO    001    001    001    001    2.0   .0    1.0   .0  28.0  2048M
+CANSP12   22    A  NO    002    002    002    002    3.0   .0    1.0   .0  27.0  1408M
+CANSP13   29    A  NO    002    002    002    002    1.0   .0    1.0   .0  29.0  1024M
+CANSP23   20    A  NO    001    001    001    001    2.0   .0    1.0   .0  28.0  512M
+CANSYSL   30    A  NO    002    002    002    002    2.0   .0    1.0   .0  28.0  1536M
+RALHCD                    D                                N/A
+TIVMVS18                    D                                N/A
+PHYSICAL   00
+
+-----
+TOTAL CP              102
+
+
+RALVMQ    07    A  NO              .0   .0  2.0   .0  7.0  73728M
+CANSP12   22    A  NO    002    002    002    002    1.0   .0    .0    .0  8.0  1408M
+CANSYSL   30    A  NO    002    002    002    002    1.0   .0    .0    .0  8.0  1536M
+PHYSICAL   00
+
+-----
+TOTAL ICF              4
+
+-----
+TOTAL ICF              76672M

```

Figure 49. Typical output for YLPAR command

The current logical partition is marked with an asterisk (*) next to its name. The line containing the logical partition name PHYSICAL displays only LPAR management overhead which cannot be attributed to a particular partition.

LPAR takes two cycles to begin displaying data. One cycle is required for initialization. If you are in TSO mode, enter the command (first cycle), then press Enter again (second cycle). Any subsequent cycles will update the data with current values.

Paging immediate commands--multiplot

OMEGAMON monitors the overall paging of the system as well as paging at the address space level. Page-ins are strictly on demand, that is, through page faults. Page-outs result from real storage management attempting to maintain an available frame list. OMEGAMON alters the high end of the scale these commands produce to the highest page rate of any address space or page set they sample. The SRM block-paging statistic counts page faults instead of page-ins. A page fault can represent the page-in of a single page or the page-in of a block of pages.

MPGI nn

Displays page-ins by address space and page data set greater nn page-ins/second. The display output is based on the page fault rate for address spaces and page data sets.

Type:

Immediate

```
MPGI03|__Task__ Pg/s 0_4_8_160|__PG N/A in goal mode_|Volume Pg/s 0_4_8_160|
+|T987R 16.1 ->. . . .| |MVS301 .0 >. . . .|
+|Tot Dmd: 65.8 ---->. .| |MVS302 .0 >. . . .|
+| | | |DLIB01 19.5 ->. . . .|
+| | | |MVS303 46.3 --->. . .|
```

Task

Shows paging by task or address space. The display includes all tasks whose page-in rate exceeds the specified threshold nn (03 in the example). It also contains a Tot Dmd field that displays the total page-in rate for the entire system. Page-in rates for tasks do not include any page transfers that occur due to swapping operations.

Volume

Shows total paging system-wide. This zone shows page-ins and page-outs. If any one page data set exceeds the threshold, all page data sets appear. Since paging due to swapping is usually a large component of the paging load, the first two zones rarely agree with the third, even if you specified MPGI00.

MPAG nn

Displays address spaces with paging rates greater than nn pages/second, and all page data sets if any are greater than nn pages/second).

Type:

Immediate

MPAG nn differs from MPGI nn in that it displays a total of both page-ins *and* page-outs. The following is an example using 01 and 00 as the threshold.

```
MPAG01|__Task__ Pf/s 0_____10|__PG N/A in goal mode_|Volume Pf/s 0_____10|
+|L3IAC5W8 1.5 >. . . .| | | |
+|L3IAI5W8 1.6 >. . . .| | | |
+|Tot Pag: 4.5 --->. . .| | | |

MPAG00|__Task__ Pf/s 0_____10|__PG N/A in goal mode_|Volume Pf/s 0_____10|
+|P3SR00VS .4 >. . . .| |PAG220 .0 >. . . .|
+|BTSR22AS .1 >. . . .| |PAG220 .0 >. . . .|
+|BTDMGRS .6 >. . . .| |PAG220 .3 >. . . .|
+|L3IAC5W8 1.5 >. . . .| |PAG222 .4 >. . . .|
+|L3IAI5W8 1.6 >. . . .| |PAG223 .2 >. . . .|
+|L3ITDSW8 .1 >. . . .| |PAG224 .3 >. . . .|
+|Tot Pag: 4.5 --->. . .| |PAG225 .3 >. . . .|
+| | | |PAG226 .3 >. . . .|
+| | | |PAG227 .3 >. . . .|
+| | | |PAG228 .4 >. . . .|
+| | | |PAG229 .3 >. . . .|
+| | | |PAG22A .3 >. . . .|
+| | | |PAG22B .1 >. . . .|
+| | | |PAG22C .3 >. . . .|
+| | | |PAG22D .4 >. . . .|
+| | | |TOTAL: 4.5 --->. . .|
```

Page data set information

The PART major command selects page data sets, and its minor commands display information about them.

Page data set information major command

PART

Selects page data sets.

Type:

Major

PART displays only the first (high-level) qualifier of the data set name. For example, for the data set SYS1.VSYS21.PLPA, only SYS1 appears. Use minor commands PAR2 through PAR9 to display additional qualifiers. For example, PAR2 displays the second qualifier of the page data set name, PAR3 displays the third qualifier, PAR4 displays the fourth qualifier, and so on.

Page data set information minor commands

PADR

Displays unit address.

PAER

Displays error count.

PANI

Displays number of IORBs built.

PAR2

Displays the second qualifier of the page data set name.

PAR3

Displays the third qualifier of the page data set name.

PAR4

Displays the fourth qualifier of the page data set name.

PAR5

Displays the fifth qualifier of the page data set name.

PAR6

Displays the sixth qualifier of the page data set name.

PAR7

Displays the seventh qualifier of the page data set name.

PAR8

Displays the eighth qualifier of the page data set name.

PAR9

Displays the ninth qualifier of the page data set name.

PAS%

Displays percent full.

This is calculated as $(PASZ-PAVL)*100/PASZ$.

PAST

Displays status.

OK/VIO indicates that the local data set is OK and also accepts VIO pages. Okay indicates that the local data set is OK but does not accept VIO pages.

PASZ

Displays size of page data sets in slots.

PATM

Displays average page I/O time in milliseconds.

Under some versions of z/OS, the I/O time figures that PATM produces are not meaningful for the Common and PLPA page data sets (the first two data sets that PART and its minors display). Therefore, OMEGAMON shows the first two data sets as blanks.

PATP

Displays type of data set.

PAVL

Displays available slots.

PAVS

Displays volume serial of DASD.

PDVT

Displays device type of the page data set.

PSIO

Displays total SIOs issued for page data set.

PXFR

Displays total pages transferred for page data set.

Append a .R to this command to show the value as a rate of pages per second.

Enqueue information immediate commands

WHO

Displays all holders and requestors of a specific enqueue.

Type:

Immediate

Format:

WHO nn

The nn argument skips the first nn lines of output.

The WHO command lets you display the jobs that have requested control of a resource in three ways:

- You can display a specific major and minor enqueue list:

```
WHO SYSIEFSD,Q4
```

- You can display a specific major enqueue and *all* of its minors:

```
WHO SYSIEFSD
```

- You can display a specific major enqueue and the minor enqueues that match a pattern, using an asterisk (*) as a wild card:

```
WHO SYSIEFSD,Q*
```

selects all minor enqueues of the SYSIEFSD major enqueue that begin with Q.

If the major or minor enqueue name is in hexadecimal, precede the name with an X and enclose the characters in single quotes ('):

```
WHO SYSIEFSD,X'00002048'
```

The asterisk as a wild card works in hexadecimal format as well:

```
WHO SYSIEFSD,X'00'*
```


Note: If special characters are used in the data set name, you must enclose the data set name in single quotes.

```
>WHO  SYSIEFSD  Q4
> OWNERSHIP      TYPE      JOBNAME      SYSID
> Exclusive Owner TSO      TSOABC      GRS1
> Exclusive Waiter BATCH    PROD0684    GRS1
```

OWNERSHIP

Indicates whether the job has the enqueue with disposition share (shared) or old (exclusive), and whether the job currently has the enqueue (owner) or is waiting for the enqueue (waiter).

TYPE

Indicates whether the user is a started task (STC), time sharing user (TSO), batch job (BATCH), mount job (MOUNT), or no longer exists and did not DEQUEUE the resource (NODEQ).

JOBNAME

Shows the JOBNAME/TSOID of the user.

SYSID

Shows the system ID of the GRS member of the GRS ring. For a definition of the term ring, see the *IBM Initialization and Tuning Guide*. If the system is not part of a GRS ring, the SYSID is always NONAME.

For generic requests, a display line preceding each group of users for the specific resource shows the full major and minor enqueue names. If the generic request specifies the minor name in hexadecimal, this display line shows the full minor name in hexadecimal. Optionally, you can put an X in column 1 of the command line to require OMEGAMON to display the minor name in hexadecimal.

The resulting output display for a generic request looks like this:

```
WHO  SYSVSAM,X'00'*
> OWNERSHIP      TYPE      JOBNAME      SYSID
> Name=SYSVSAM, X'0079A00B36280C9'
> Shared Owner   STC      EPICOL      NONAME
> Shared Owner   BATCH    ACP970      NONAME
> Name=SYSVSAM, X'0079A00B36280D6'
> Shared Owner   BATCH    WHS120T     NONAME
```

XQCB

Displays enqueue conflicts and RESERVEs.

Type:

Immediate

XQCB can have an optional argument X (XQCBX) that excludes from the display any enqueues whose major name appears in the enqueue exclusion table, which the XQN immediate command defines.

A plus sign (+) after the minor name indicates that it is truncated. An S in column 1 limits the display of minor names to 43 bytes.

In a shared DASD environment, some address spaces doing I/O to a sharedDASD issue what is called a RESERVE. (Not all address spaces use the enqueue—JES2 on the checkpoint data set, for example). In such a case, XQCB also indicates any RESERVEs.

Format:

```
XQCB sooooooooo wwww ttt mmmmmmmm nnnn... RESV xxxxyvvvvvvv
```

S

* means user is swapped out.

oooooooo

Owner of the enqueue—first user listed owns resource.

wwwww

Either WAIT (user is waiting) or blank (user has resource).

ttt

Either SHR (shared) or EXC (exclusive).

mmmmmmmm

Name of major resource. If a dot appears in any position of this field, data cannot be interpreted because z/OS has unprintable data. This is not a problem with OMEGAMON.

nnnn...

Minor name of resource.

RESV

Indicates that this is a RESERVE.

xxxx

Indicates the hex address of the RESERVED device.

y

Indicates UCB reserve status (or blank):

- +** reserve pending
- *** reserved
- reserve release pending

(blank)

the device is not currently in any of the hardware reserve statuses

vvvvvv

Volser of the DASD.

If you enter the XQCB command with an argument of *nn*, the first *nn* lines of output suppress. For long displays, this option works like a scrolling feature. If you want to suppress a number of lines greater than 99, use XQC*nnn*.

To see enqueues held by a specific job, use the QCB address space minor command.

In some z/OS environments where there is a great deal of enqueue activity (particularly reserve enqueues), the enqueue chain may be reorganized while OMEGAMON is processing it. In this case, an error recovery message indicates that OMEGAMON encountered an invalid chain:

```
OB0910 PROGRAM CHECK. RECOVERY SUCCESSFUL
```

OMEGAMON continues processing remaining commands and retries this command on the next cycle.

Sometimes enqueue minor names contain unprintable characters (for example, when a hex address is part of the name). Certain common z/OS enqueues of this type have been identified, and will be converted into a more meaningful display:

```
IEZIGGV3 CAXWA=xxxxxx
SYSZTIOT JOB=ccccccc DSAB=xxxxxx
SYSVSAM CI#=xxxxxx ACB=xxxxxx STAT=c
SYSCTLG INDEX=ccccccc UNIT=xxx
SYSZIGGI JOB=ccccccc
SYSZVMV UNIT=xxx
SYSZVSRL UNIT=xxx
SYSIKJBC RBA=xxxxxxxx
```

where *c* indicates an alphanumeric and *x* is a hex digit.

It is possible that one of the resource requests may be from another system altogether. When this happens, a second display line appears to indicate the user's GRS system name. For example:

```

XQCB  PAYROLL      EXC  SYSDSN  PAYROLL.MASTER
+      GRS  SYSNAME=GRSSYS01  ENQUEUE  TYPE=SYSTEMS
+      PAYUPD  WAIT  EXC  SYSDSN  PAYROLL.MASTER

```

Here, job PAYUPD is waiting to use data set PAYROLL.MASTER because job PAYROLL already has it allocated from GRS system GRSSYS01.

Data set information immediate commands

DSN

Displays data set label information given DSN.

Type:

Immediate

DSN displays tape or disk volume data set label information. The command name begins in column 2 and the data set name begins in column 7. OMEGAMON automatically supplies the name of the volser, and replaces the command DSN with DSNV to indicate that it found the volser name.

To execute the command in dedicated mode, you must position the cursor down to the next line.

If you enter:

```
DSN OMEGA.TEST.MACLIB
```

the result is:

```

DSN OMEGA.TEST.MACLIB
>DSNV CBEAV3 OMEGA.TEST.MACLIB
>      Creat-Dt Expir-Dt Last-Acs  XT DSORG BLKSZ  LRECL RECFM
>      05/06/08 01/00/00 05/08/08   1  PO  27920   80 FB
>      Prim-Alloc Used-Alloc  Sec-Alloc
>      3060      3051      728B

```

Creat-Dt

Creation date

Expir-Dt

Expiration date

Last-Acs

Date of last access

P

Password protection

Alloc

Total tracks allocated

Used

Total tracks used

XT

Number of extents

SEC

Secondary allocation (in tracks or blocks)

DSORG

Data set organization

BLKSZ

Block size

LRECL

Logical record length

RECFM

Record format

The > next to DSNV indicates that the command changed to a comment to prevent re-executing on subsequent cycles. To allow the command to execute on the next cycle, space over the >.

If you want information on a data set that is uncataloged, you must use DSNV instead of DSN, and provide the name of the volser along with the data set name.

LOC

Displays all users of a specific data set.

Type:

Immediate

Format:

LOCnnn

The *nnn* argument skips the first *nn* lines of output.

LOC displays all address spaces that currently have the data set allocated (either with DISP=SHR or DISP=OLD), via normal job scheduling or dynamic allocation appear.

The format of LOC is:

```
LOC    dataset name
```

If special characters are used in the data set name, enclose the data set name in single quotes (').

If you enter:

```
LOC    TDNY01.CNTL
```

the result is:

```
>LOC TDNY01.CNTL TS0002, 3375
> OWNERSHIP      TYPE      JOBNAME      SYSID
> Exclusive Owner TSO      TDNY01      A430
> Shared waiter  STC      RMFA        B430
> Shared waiter  TSO      TDOM02      B430
> Shared waiter  BATCH   TDOM01A     C083
```

OWNERSHIP

Indicates whether the job has the data set allocated with disposition SHR (shared) or OLD (exclusive), and whether the job currently has the data set allocated (owner) or is waiting in allocation for the data set (waiter).

TYPE

Indicates whether the user is a started task (STC), time sharing user (TSO), batch job (BATCH), mount job (MOUNT), or no longer exists and did not DEQUEUE the resource (NODEQ).

JOBNAME

Displays the jobname/TSOID of the user.

SYSID

Shows the system ID of the GRS member of the GRS ring. If the system is not a part of a GRS ring, the system ID is always NONAME.

Control block displays

At times it is essential to explore z/OS control blocks in detail. The DUMP major command and its minor commands let you explore individual z/OS control blocks, each minor command displaying a different control block.

Control block display major command

DUMP

Displays dumps of control blocks as specified by its minors.

Type:

Major

An argument of AD shows the starting address of the control block.

To dump a UCB, use the DVMP minor command of DEV.

For more information about devices, see [Chapter 9, “Device information,”](#) on page 185.

Control block display minor commands

By default the display is in hexadecimal and character format. To view the display only in character, add a C in column 1.

ASCBnn

Dumps address space control block (ASCB) for ASID *nn*.

Type:

Minor of DUMP

The variable *nn* is a decimal number.

Use the ASC*nnn* command to dump the ASCB for ASID *nnn*.

ASMV

Dumps Auxiliary Storage Manager Vector Table.

Type:

Minor of DUMP

ASVT

Dumps Address Space Vector Table.

Type:

Minor of DUMP

ASXBnn

Dumps address space extension block (ASXB) for ASID *nn*.

Type:

Minor of DUMP

nn is a decimal number.

Use the ASX*nnn* command to dump the address space extension block for ASID *nnn*.

This command only works for the user's own ASID and for ASID 01 unless OMEGAMON runs APF authorized. This is because the ASXB resides in private storage, and OMEGAMON has to schedule an SRB to get to it.

CCT

Dumps SRM CPU Management Control Table.

Type:

Minor of DUMP

CMBxxx

Dumps Channel Measurement Block for device xxx.

Type:

Minor of DUMP

CMCT

Dumps Channel Measurement Control Table.

Type:

Minor of DUMP

CPMT

Dumps Channel Path Measurement Table.

Type:

Minor of DUMP

CSD

Dumps Common System Data area.

Type:

Minor of DUMP

CVT

Dumps Communications Vector Table.

Type:

Minor of DUMP

GDA

Dumps Global Data Area.

Type:

Minor of DUMP

GVT

Dumps GRS Vector Table.

Type:

Minor of DUMP

ICHP

Dumps installed channel path table.

Type:

Minor of DUMP

ICT

Dumps SRM I/O Management Control Table.

Type:

Minor of DUMP

LCCAnn

Dumps logical configuration communication area (LCCA) *nn*.

Type:

Minor of DUMP

OMEGAMON must be APF authorized for the LCCA command to work.

LSCT

Dumps logical swap control table.

Type:

Minor of DUMP

MCT

Dumps SRM Storage Management Control Table.

Type:
Minor of DUMP

OUCBnn
Dumps SRM UCB for ASID *nn*.

Type:
Minor of DUMP

The variable *nn* is a decimal number.

Use the OUC*nnn* command to dump the SRM user control block for ASID *nnn*.

OUXBnn
Dumps SRM user extension block for ASID *nn*.

Type:
Minor of DUMP

The variable *nn* is a decimal number.

Use the OUX*nnn* command to dump the SRM user extension block for ASID *nnn*.

PAREnn
Dumps ASM paging activity reference table entry *nn*.

Type:
Minor of DUMP

To display the entire table, use PART.

The variable *nn* is a decimal number.

PART
Dumps ASM paging activity reference table.

Type:
Minor of DUMP

To display one entry in the table, use PARE*nn*.

PCCAnn
Dumps physical configuration communication area *nn*.

Type:
Minor of DUMP

PSAn
Dumps Prefixed Storage Area for the processor specified.

Type:
Minor of DUMP

n is the number of the processor for which you want to see the PSA.

Note: Multiprocessors use a technique called storage prefixing to access the PSA. When you use the PSAn command to dump the PSA of the processor on which OMEGAMON is running, you see the contents of the SYSGEN PSA, which might not match what you see when you dump location 0 using the MLST command.

PVT
Dumps RSM Paging Vector Table.

Type:
Minor of DUMP

RABC
Dumps RSM address space block for common area.

Type:
Minor of DUMP

Use the `RABnnn` command to dump the RSM address space block for ASID `nnn`.

RCE

Dumps RSM Control and Enumeration area.

Type:

Minor of DUMP

RCT

Dumps SRM Resource Control Table.

Type:

Minor of DUMP

RIT

Dumps RSM Internal Table.

Type:

Minor of DUMP

RMCA

Dumps SRM Control Area.

Type:

Minor of DUMP

RMCT

Dumps SRM Control Table.

Type:

Minor of DUMP

RMPT

Dumps SRM Parameter Table.

Type:

Minor of DUMP

SAREnn

Dumps Swap Activity Reference Table entry `nn`.

Type:

Minor of DUMP

The variable `nn` is a decimal number.

To display the entire table, use SART.

SART

Dumps Swap Activity Reference Table.

Type:

Minor of DUMP

To display a single entry in the table, use `SAREnn`.

SCVT

Dumps Secondary Communication Vector Table.

Type:

Minor of DUMP

SPL

Dumps Service Priority List.

Type:

Minor of DUMP

WMST

Dumps SRM Workload Manager Specification Table.

Type:

Minor of DUMP

The following figure shows typical hex dumps of address space control block 19 and the SRM control table.

```

DUMP
ASCB19          <Dump ASID 19 TEST23>                               Addr=00FCC9A8
+ 0000 C1E2C3C2 00FDC130 0000BC10 00000000 *ASCB..A.....*
+ 0010 0081D248 000000A0 00000000 00296310 *.aK.....*
+ 0020 00000000 00030002 000001FF 00000000 *.....*
+ 0030 0081F548 C0FE0260 00FE8020 00000000 *.a5.{.-.....*
+ 0040 00000000 1706F000 94FC0D1B 09A2D000 *......0.m....s}*
+ 0050 00000000 8081D7E0 94FC0CBC 00000000 *.....aP\m.....*
+ 0060 0081CD90 FFFF0100 00000000 0081F300 *.a.....a3.*
+ 0070 002B0044 00000000 ... <dump of ASCB> ...
=====
RMCT                               Addr=000281F8
+ 0000 D9D4C3E3 00028318 000283C8 00028428 *RMCT..c...cH..d.*
+ 0010 00028610 00028678 00FE6220 00FEE978 *..f...f.....Z.*
+ 0020 0002D278 00000000 000287A0 000287F8 *..K.....g...g8*
+ 0030 00028B78 00028B38 00028A68 00028760 *.....g-*
+ 0040 00031923 00024D94 0004F58C 00028E78 *.....(m..5.....*
+ 0050 00028EAC 00000000 00FEA678 00028BA8 *.....w....y*
+ 0060 0000A466 00028C10 00028BC8 00028BE0 *..u.....H...\*
+ 0070 00028BF8 00028898 0FCCC5F7 ... <dump of RMCT> ...
=====

```

LPA modules immediate command

FMOD

Finds entry point of module in link pack area.

Type:

Immediate

FMOD locates and determines the status of modules in the Link Pack Area (LPA). You supply the module name. The output for FMOD is as follows:

```

FMOD module-name *ACT* aaaa Entry=xxxxxx Load=xxxxxx Use Cnt= ddd
+ cccc Address=xxxxxx Major Module=mmmmmmmm

```

ACT

Appears if the module-name was found on the Active Link Pack Area Queue (ALPAQ). This means that a CDE was found for the module.

aaaa

Location of the module:

FLPA

If the module is located in the Fixed Link Pack Area.

MLPA

If the module is located in the Modified Link Pack Area.

PLPA

If the module is located in the Pageable Link Pack Area.

EFLPA

If the module is located in the Extended Fixed Link Pack Area.

EMLPA

If the module is located in the Extended Modified Link Pack Area.

EPLPA

If the module is located in the Extended Pageable Link Pack Area.

Note: The extended areas (that is, the areas above the 16M line) are prefixed by an E. For example, the Extended PLPA would be indicated by EPLPA.

Entry=

The entry address (xxxxxx represents the hex address) for this module.

Load=

The starting address of the load module in which this module name resides. Most of the time the Entry address will be the same as the Load address. It can be different if the first instruction to be executed in this load module is not at offset 0.

Note: The load address is *not* given for a minor CDE/LPDE. To find this information, enter FMOD for the major module.

Use Cnt=

The number of requests currently active for the specified module. The use count is always 0 for minor CDEs and for LPDEs; therefore, it is not shown for these entries. For modules in either the MLPA or the FLPA, the use count is initialized to 1 at IPL. For modules in the PLPA, the use count does not appear unless the module is actually being requested. In such a case, the first request sets the use count to 1. Module aliases have no use count.

Note: The use count goes up as requests are made for the module and goes down when requests for the module become inactive.

cccc Address=

cccc is either CDE or LPDE. The address represents the location of the control block used to obtain the data shown.

Major Module=

The name of the module for which the requested module is a minor (an alias).

```
FMOD IKJEFT01 *Act* PLPA  Entry=BAF000  Load=BAF000  Use Cnt= 37
+    CDE  Address=FDFA30
FMOD IKJEFT02          PLPA  Entry=BA9000  Load=BA9000  Use Cnt= 76
+    CDE  Address=FAA4F0
```

In this example, module IKJEFT01 is currently active, begins at location BAF000, and has a use count of 37. CDE address is FDFA30. Module IKJEFT02 is loaded at address BA9000, but is not in use at the moment. CDE address is FAA4F0.

GRS immediate command

GRS

Displays global resource serialization information.

Type:

Immediate

Global Resource Serialization (GRS) manages the sharing of data sets and other resources between separate systems by connecting the systems into a communication's ring using channel-to-channel adapters. If any link in this ring fails, the entire ring deactivates until the system or the operator restarts it.

The XGRS exception reports when the GRS ring deactivates.

```
GRS  SYSNAME=xxxxxxx      GRS Version=vvvv      Status=ssss
+    Exclusion RNL=000000  Inclusion RNL=000000  Conversion RNL=000000
```

Workload Manager

The WLM major command is the data collector for the function. It is used with the SDPD and SRVC minor commands. The command collects a new set of performance data each time the Enter key is pressed.

Other than a summary line displaying the current service definition and policy names, the WLM command does not display any data itself.

Use the SDPD minor command to display details about the currently active WLM service definition and service policy. Use the SRVC minor command to display details about all service classes and all periods for a single service class, or a single service class period.

WLM

Initializes collection of Workload Manager data and displays WLM service definition and policy names.

Type:

Major

```

----- ZADV      VTM      OM/DEX  V420./C SYS  03/05/09 14:07:26
WLM >> Service Definition : prod      Service Policy : FEB0502
sdpd
+
+Service Definition : prod
+Description       : Production Goal Mode
+Installed         : 02/03/09 at 19:47:00 by WANGCM   on SP22
+
+Service Policy    : FEB0502
+Description       : TSO, STC, STCCMS changes
+Installed         : 02/03/09 at 19:47:04 by WANGCM   on SP22
  srvc
+
+Service
+Class   Prd  Goal  Imp  Goal  Dur  Perf  Avg  Actual  Avg  Resource
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
+BATCH   1  Velo  3    30   700  0.00          203K  BATCH
+BATCH   2  Velo  3    15           0.00          BATCH
+BATHI   1  Velo  3    21           0.00          BATCH
+BATPROD 1  Velo  2    20           0.00          BATCH
+BATSOAKR 1  Velo  3    10  51200  0.00          BATCH
+BATSOAKR 2  Velo  4     5           0.00          BATCH
+CICS    1  PctR  2   1000  0.50          500ms  CICS
+CICSCEMT 1  PctR  2   2000  0.00          CICS
+CICSRAID 1  PctR  2   1000  0.00          CICS
+CICSTEST 1  PctR  2   1000  0.50          500ms  CICS
+CICSVSAM 1  PctR  2   1000  0.00          CICS
+ENCLAVES 1  AvgR  2  10000  0.00          NCLV
+IMSCL1  1  PctR  2    400  0.00          IMS
+IMSCL2  1  AvgR  2    600  0.00          IMS
+IMSCL4  1  AvgR  2   1000  0.00          IMS
+IMSMPRS  1  Velo  2    30   0.00          BATCH
+OMVSDAEM 1  Velo  2    30   0.00          OPENEDTN
+OMVSJOBS 1  Velo  1    20  1000  0.00          148  OPENEDTN
+OMVSJOBS 2  Velo  2    20  5000  0.00          OPENEDTN
+OMVSJOBS 3  Velo  3    10   0.00          OPENEDTN
+OMVSKERN 1  Velo  2    30   0.00          OPENEDTN
+SLEEP    1  Disc           0.00          STC      SLEEP
+STC      1  Velo  2    30  5000  0.30          100%   376K  STC
+STC      2  Velo  3    20           0.34  129841  59%  1130K  STC
+STCCMS   1  Velo  2    30  55000  0.00          2718K  STC
+STCCMS   2  Velo  3    20           0.00          STC
+STCHI    1  Velo  2    30   0.00          STC
+STCMED   1  Velo  3    20   0.00          STC
+STCMQ    1  Velo  3    20   0.00          STC
+STCONLN  1  Velo  2    20   0.20          100%   153K  STC
+STCPROD  1  Velo  2    20   0.00          536K  STC
+TIMEOUT  1  Disc           0.00          STC      TIMEOUT
+TSO      1  AvgR  1    500  1000  0.17          85    85ms  126K  TSO
+TSO      2  PctR  1    750   200  0.00          TSO
+TSO      3  AvgR  2   5000  5000  0.00          TSO
+TSO      4  Velo  3    30   0.00          TSO
+TSOHI    1  AvgR  1    500  1200  0.00          TSO      TSOHI
+TSOHI    2  AvgR  1   1000  3000  0.00          TSO      TSOHI
+TSOHI    3  AvgR  1   4000  12000  0.00          TSO      TSOHI
+TSOHI    4  Velo  3    30   0.00          TSO      TSOHI
+TWOMIP   1  Disc           0.00          STC      TWOMIP
+SYSTEM   1  Syst           0.00          347K  SYSTEM
+SYSSTC   1  Syst           0.00          216K  SYSTEM
+SYSSTC1  1  Syst           0.00          SYSTEM
+SYSSTC2  1  Syst           0.00          SYSTEM
+SYSSTC3  1  Syst           0.00          SYSTEM
+SYSSTC4  1  Syst           0.00          SYSTEM
+SYSSTC5  1  Syst           0.00          SYSTEM
+SYSOTHER 1  Syst           0.00          SYSTEM

```

Figure 50. Output for WLM with SDPD and SRVC minor commands

SRVC

Display details about all service classes and all periods for a single service class, or a single service class period.

Type:
Minor of WLM

Format:
[n]SRVC [classname]where

blank (in the first column)

Displays a single line summary for all periods within all service classes.

n

Displays detail information for the selected period within the specified service class. Requires that a service class name be specified.

classname

Displays a single line summary for all periods for the specified service class.

```

----- ZMENU      VTM      OM/DEX  V420./C SYS  03/05/09 14:14:37
> Help/News PF1  Exit PF3   Keys PF5   Colors PF18  Cmd mode PF24
WLM  >> Service Definition : prod      Service Policy : FEB0502
1srvc BATCH
+
+Workload Name      : BATCH      Description : Batch Jobs
+Service Class Name : BATCH      Description : Batch Jobs
+Service Class Period : 1          Performance index : 0.00
+Importance         : 3 (Medium)
+Goal Type           : Velocity > 30%
+Duration            : 700 (Service Units)
+Actual              : 0
+Average Response Time : 0
+CPU %               : 0.00
+IFA %               : 0.00
+IFA on CP %         : 0.00
+zIIP %              : 0.00
+zIIP on CP %        : 0.00
+Average Storage     : 0

```

Figure 51. Output for SRVC with period and service class name specified

SDPD

Displays service definition and service policy information.

Type:

Minor of WLM

Use the SDPD command to display details about the currently active WLM service definition and service policy for one or more service classes and periods.

Chapter 8. CSA Analyzer

The CSA (Common Storage Area) Analyzer is an OMEGAMON feature that monitors and analyzes common storage usage and growth in the following areas: common service area (CSA), extended common service area (ECSA), system queue area (SQA), and extended service queue area (ESQA).

The CSA Analyzer generates displays on overall common storage usage, high level graphic displays on usage trends, and reports on growth at the job or system level. It can generate detailed, itemized reports on outstanding getmain requests, identifying the user, the getmain area, and the return address of the getmain request. The CSA Analyzer also identifies users of a particular block of common storage.

Included within the CSA Analyzer is a recovery facility that identifies and frees Common Storage Area obtained by a job that is no longer active.

Configuring the CSA Analyzer

Configure the z/OS system initialization parmlib DIAGxx member to specify what areas of common storage should be tracked by the z/OS VSM common storage tracking function. There are two keyword control specifications for the VSM Common Storage Tracking that affect the CSA Analyzer:

CSA (ON|OFF): Tracking common service areas (CSA & ECSA) status

SQA (ON|OFF): Tracking system queue areas (SQA & ESQA) status

If both CSA and SQA controls are "OFF", the CSA Analyzer does not provide any common storage tracking information.

Changing the startup parameters

Initially, use the default values for the startup parameters. If you need to change the values, edit the sample CSA Analyzer PROC that is in *rhilev.midlev.RKANSAMU*.

The following table describes the CSA Analyzer startup options and default values:

Parameter	Symbolic	Default value
Reserved number of bytes of unused ECSA	RESV= <i>nnnnnn</i>	0
Target sampling accuracy percentage	SAMP= <i>nnn</i>	100
Region size	REG= <i>nnnn</i>	2M
CPU time limit	TIM= <i>nnnn</i>	1439
Dispatching priority	PRTY=' <i>(xx,yy)</i> '	'(15,15)'
System trend interval	SINT= <i>nn</i>	15
Number of system trend buckets	SNUM= <i>nnn</i>	96
Job trend interval	JINY= <i>nn</i>	5
Number of job trend buckets	JNUM= <i>nnn</i>	6

For more information about these parameters, see [“Startup parameter symbolics” on page 178](#)

Startup parameter symbolics

This section describes the startup parameter symbolics.

RESV=nnn

Reserved number of bytes of ECSA that are unused by the CSA Analyzer. This reserved space is a storage cushion that can be used by any task other than the CSAA. If the CSAA cannot obtain ECSA storage while leaving this reserved space, message CSAA804E is issued and the CSAA suspends its data collection.

SAMP=nnn

Target sampling accuracy percentage. You can use this parameter to reduce the CPU processing used in sampling common storage consumption in exchange for a lower degree of accuracy.

If you specify SAMP=100, sampling occurs once a second.

If you specify a value less than 100, the CSA Analyzer does not sample if the summary allocation totals for each common storage area are within the sampling percent of the values in the Global Data Area (GDA).

The valid range of values is 95 to 100.

REG=nnnn

Size of the region in which the CSA Analyzer address space runs.

TIM=nnnn

CPU time limit for the CSA Analyzer. A low value might prematurely terminate the CSA Analyzer subsystem.

PRTY='(xx,yy)'

Dispatching priority for the CSA Analyzer. If this value is too low, the CSA Analyzer might not be able to collect data on all events. The ICS and IPS values might override this value.

SINT=nn

Interval length (in minutes) between sampling of common storage used for system trending information. Adjust this parameter in conjunction with the SNUM= parameter to show meaningful system trending information for your installation. The valid range of values is 1 to 60.

SNUM=nnn

Number of sample buckets to keep for system trending. To turn off system trending, specify SNUM=0. Each sample bucket uses 56 bytes. The valid range of values is 0 to 999.

JINT=nn

Interval length in minutes, between the sampling of common storage used for job trending information. Change this parameter in conjunction with the JNUM= parameter to show meaningful job trending information for your installation. The valid range of values is 1 to 60.

JNUM=nnn

Number of sample buckets to keep for job trending. To turn off job trending, specify JNUM=0. Each sample bucket uses 16 bytes for each job with allocated common storage. The valid range of values is 0 to 999.

Implementing CSA Analyzer IPCS support

The CSA Analyzer provides an IPCS verb exit that enables you to view CSA Analyzer data in an MVS system dump. The IPCS verb exit is a load module in *rhilev.midlev.RKANMOD(KCSIPCS)*. The KCSIPCS load module must be in a load module library that is available to IPCS, such as a step library, job library, or a link library.

To access the KCSIPCS module, issue the following command from an IPCS session:

```
VERBX KCSIPCS 'parm'
```

where *parm* is one of these parameters:

SUMMARY

Produces output similar to that of the OMEGAMON CSAA USAGE and CSAA SUMMARY commands. This parameter is the default.

DETAIL

Produces all the information that is provided by the SUMMARY parameter, plus the storage block information that is produced by the CSAA DETAIL command.

STORAGE

Produces all the information that is provided by the DETAIL parameter, plus the first 16 bytes of each GETMAINed element, in hexadecimal and character format.

If you want to run IPCS as a batch job, put a VERBX entry in the appropriate IPCS control member of SYS1.PARMLIB.

CSA Analyzer commands

The CSA Analyzer provides CSA usage information through the OMEGAMON CSAA and CSAF immediate commands.

CSAA immediate command

The CSAA immediate command displays common storage area information.

Type:

Immediate

Format:

CSAA [function keyword] (filter keyword)

Table 9 on page 179 shows the available function keywords and their associated filter keywords.

The CSAA command entered with only a function keyword produces a display based on the defaults for the filter keywords. The full command syntax is included in the display.

Function keyword	Definition	Applicable filter keyword
SUMMARY	Shows overall common storage utilization. You can use this command to determine if there is a common storage shortage problem in any of the four common storage areas.	none
TREND	Displays the utilization trend of CSA, ECSA, SQA, or ESQA at the system or job level.	AREA, SYSTEM or JOB, ASID
USAGE	Lists the users of CSA, ECSA, SQA, or ESQA.	AREA, BOUND
GROWTH	Displays the fastest growing users of CSA, ECSA, SQA, or ESQA.	AREA, BOUND
DETAIL	Generates a detailed report on outstanding GETMAINed items. You must specify a jobname.	AREA, ASID, SUBPOOL, MINSIZE, JOB, KEY
ORPHAN	Displays outstanding GETMAINed items whose original requestors are no longer active. You can specify SYSTEM for all jobs, or the jobname for a specific job.	AREA, ASID, SUBPOOL, MINSIZE, JOB or SYSTEM, KEY
OWNER	Shows the original requestor of a block of common storage. You must specify the start address or the address range of the common storage block.	RANGE, KEY

Table 10 on page 180 describes the CSAA filter keywords used with the function keywords to help tailor the display.

Filter Keyword	Definition	Default
AREA(c)	CSA, ESCA, SQA, or ESQA	CSA
SYSTEM	Generates a system level report.	SYSTEM
JOB(c)	Generates a report for a specified job; an * indicates all jobs.	n/a
ASID(n)	ASID; an * indicates all ASIDs of the specified jobname.	*
BOUND(n)	Specifies a lower bound in percent to limit display.	0
SUBPOOL(n)	Subpool number; an * indicates all subpool numbers within the specified area.	*
KEY(n)	Storage key; any number from 0 to 15 or *. An * indicates all storage keys.	*
MINSIZE(n)	Limits the display to areas whose size is greater than or equal to minsize.	0
RANGE(l,h)	Specifies low and high addresses that indicate the desired address range.	all common storage addresses

The CSAA command retains values for its keywords; therefore, you do not need to re-enter their values each time you execute the command. Also, it displays the current values with the keywords so that they can be changed easily.

Note: An asterisk (*) to the right of a jobname indicate that the job has terminated.

A jobname of *SYSTEM* indicates that the issuer of a request is unknown.

CSAA command examples

CSAA command output supports extended color. Exception thresholds trigger highlighting where applicable.

Figure 52 on page 180 shows a display of CSAA USAGE using the AREA and BOUND filter keywords.

```

CSAA  USAGE  AREA(CSA)  BOUND(1)
+  JOBNAME  ASID      USAGE          CSA USAGE BY JOB
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+SXSMI22C  0028    1285K  22.1%|----->
+TESTJOB1  0013    124K   2.1%|
+TESTJOB2  000C     98K   1.7%|
+SXSMI22D  0029     75K   1.3%|

```

Figure 52. CSAA USAGE sample output

Figure 53 on page 181 is a display of CSAA TREND with the AREA and SYSTEM filter keywords.

CSAA	TREND	AREA(CSA)	SYSTEM	CSA USAGE TRENDS FOR SYSTEM										
+	TIME	USAGE		0	1	2	3	4	5	6	7	8	9	100%
+ 7:45	08/03	1913K	33.0%	----->										
+ 6:43		1915K	33.0%	----->										
+ 6:28		502K	8.7%	---->										
+ 6:13		500K	8.6%	---->										
+ 5:58		478K	8.2%	---->										
+ 5:43		1887K	32.5%	----->										
+ 5:28		1887K	32.5%	----->										
+ 5:13		472K	8.1%	---->										
+ 4:58		471K	8.1%	---->										

Figure 53. CSAA TREND sample output

Figure 54 on page 181 shows CSAA OWNER output using the RANGE filter keyword.

CSAA	OWNER	RANGE (B55000, B66FFF)		KEY (*)								
+		ASID	SIZE	START	END	FIX	SP	KEY	AGE	RETURN		
+ SXSMI22C	0028	8192	00B55000	00B56FFF	YES	228	7	1:17	HR	80C11602		
+ SXSMI22C	0028	4096	00B57000	00B57FFF	NO	241	7	1:17	HR	80C1161E		
+ SXSMI22C	0028	4096	00B5C000	00B5CFFF	NO	231	7	1:17	HR	80C11602		
+ SXSMI22C	0028	4096	00B5E000	00B5EFFF	NO	231	7	1:17	HR	80C11602		
+ SXSMI22C	0028	4096	00B60000	00B60FFF	NO	231	7	1:17	HR	80C11602		
+ SXSMI22C	0028	4096	00B62000	00B62FFF	NO	231	7	1:17	HR	80C11602		
+ SXSMI22C	0028	4096	00B66000	00B66FFF	YES	228	7	1:17	HR	80C11602		

Figure 54. CSAA OWNER sample output

FIX

Indicates whether or not the subpool is fixed.

SP

The subpool.

RETURN

The return address of the getmain requestor. It points to the instruction after the getmain request.

CSAF immediate command

Displays and frees orphan storage blocks.

Type:

Immediate

You can specify SYSTEM for all jobs, or the jobname for a specific job. Place an F before the job to free the displayed orphan storage areas.

The OMEGAMON CSAF command uses the following filter keywords:

- AREA
- JOB or SYSTEM
- ASID
- MINSIZE
- SUBPOOL
- KEY

Important: Orphaned storage is all storage requested by an address space that has since terminated. Other users can still reference this storage. You therefore should not attempt to free any storage unless you know how the storage has been used and are sure it is no longer needed. If you free storage that other users are currently referencing, those users may abnormally terminate.

Some CSAA commands display a RETURN field. This is the return address of the getmain requestor. It points to the instruction after the getmain request.

CSAF command example

Figure 55 on page 182 shows CSAF output.

```
CSAF AREA(CSA)  SYSTEM
:   SUBPOOL(*)  KEY(*)  MINSIZE(0)
+
+   ENTER 'F' IN FRONT OF JOBNAME TO FREEMAIN THE ORPHAN STORAGE AREA.
+   >>> THIS FREEMAIN FACILITY SHOULD BE USED WITH EXTREME CAUTION <<<
+
+   ASID  SIZE  START  END  FIX  SP  KEY  AGE  RETURN
:  _TDSMS06 * 01F6   320 00C1DC40 00C1DD7F NO 241  8  2:09 HR 40000000
:  _SXSMI22C * 0022   152 00BDD048 00BDD0DF NO 231  7  5:44 HR 40000000
:  _$IAMSTR * 000D  4920 00C59CC8 00C5AFFE NO 241  0  7:01 HR 400078D0
:  _$IAMSTR * 000D  1176 00C5EB68 00C5EFFF NO 241  0  7:01 HR 400078D0
```

Figure 55. CSAF sample output

CSA Analyzer-related exceptions

Three groups of exceptions are available to warn of any problems detected with the CSA Analyzer: job usage exceptions, job growth exceptions, and system growth exceptions.

Job usage exception

XUCA

Indicates any job's excessive ECSA usage.

XUCB

Indicates any job's excessive CSA usage.

XUSA

Indicates any job's excessive ESQA usage.

XUSB

Indicates any job's excessive SQA usage.

Job growth exceptions

XJCA

Indicates any job's excessive usage growth of ECSA over a time interval.

XJCB

Indicates any job's excessive usage growth of CSA over a time interval.

XJSA

Indicates any job's excessive usage growth of ESQA over a time interval.

XJSB

Indicates any job's excessive usage growth of SQA over a time interval.

System growth exceptions

XSCA

Indicates the system's excessive usage growth of ECSA over a time interval.

XSCB

Indicates the system's excessive usage growth of CSA over a time interval.

XSSA

Indicates the system's excessive usage growth of ESQA over a time interval.

XSSB

Indicates the system's excessive usage growth of SQA over a time interval.

CSA Analyzer-related minor commands

All OMEGAMON for MVS address space major commands now support the CSA Analyzer with four minor commands.

CSA

Displays a job's usage of CSA.

ECSA

Displays a job's usage of ECSA.

ESQA

Displays a job's usage of ESQA.

SQA

Displays a job's usage of SQA.

Chapter 9. Device information

Device information consists of these areas:

- Generic Device Information
- DASD Information
- Tape Information
- Long-Term Device Utilization
- Channel Information

Disk major commands display the unit addresses of the disks they select.

Device listing commands

These immediate commands display lists of devices.

Command	Description
DLST	Lists all online and offline disks.

Type:

Immediate

A hyphen (-) in the command display indicates the selected offline disks.

Command	Description
GLST	Lists esoteric and generic device names by device class.

Type:

Immediate

Format:

```
GLST <cccc|cccc &concat;>
```

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:

```
-----  
GLST  
+ TAPE DEVICES:  
+   3480X   3400-6   TAPE   CART   T3480   3480   3400-9   3400-5  
+   3400-3  
+  
+ COMMUNICATIONS DEVICES:  
+   3705
```

Use of the optional operands is limited to MVS 4.1 and later.

See also the GDEV command.

Command	Description
TLST	Selects all tape drives (online/offline).

Type:

Immediate

A hyphen (-) in the command display indicates the offline tape drives selected.

.DMP command

Command	Description
.DMP	Displays information at hex offset xx from ASCB or Unit Control Block (UCB).

Type:

Minor of device majors

Format:

c .DMPxx

or .DMxxx

If you place the letter C in column 1, .DMP dumps 8 characters at offset xx in character format. If you want to specify a three-digit offset, use .DMxxx.

This command is considered to be a generalized minor, in that it applies to more than one group of major commands.

When you use .DMPxx with an address space major command, it displays 4 hex bytes at offset xx from the ASCB or the UCB.

When you use .DMPxx with a device major command, it displays 4 hex bytes at offset xx from the device's UCB.

Disk information

This section describes major commands which select disks, the immediate command that displays the space available on disks, and the minor commands which display detailed information about the disks.

Disk select major commands

Command	Description
DEV	Selects a disk with volser cccccc or address xxx.

Type:

Major

Format:

DEV cccccc|xxx

If you supply the volser or the address of a disk, DEV displays the volser, the address, and the online or offline status of the disk. If you enter:

```
DEV 520
```

the result is:

```
DEV 520 volser=TS0099 Online Alloc
```

Alternatively, you could supply the volser (TSO099).

For PAV (parallel access volume) devices, the word "Base" appears before "Volser" in the display.

Command	Description
DEVL	Selects list of online disks by volser cccccc or unit address xxx.

Type:

Major

Format:

DEVL cccccc ... cccccc|xxx ... xxx

DEVL selects a list of disks for examination, whereas DEV selects only *one* device at a time.

The following example shows disks at addresses 123 and 141 and volsers TSO021 and TSO022:

```
DEVL 123 TS0021 TS0022 141
```

If you list an invalid address or volser, or if you specify the same disk twice, OMEGAMON eliminates the invalid or duplicate value from the list.

Command	Description
DEVP	Selects a list of online disks using patterns set with .SPT.

Type:

Major

Format:

DEVP/n

The variable *n* is a number from 0 to 9 indicating the pattern set with the .SPT immediate command.

For example, you could set the pattern and then display all online disks beginning with TSO as shown in this figure:

```
.SPT/9   TSO*
DEVP/9   TSO024  TS0025  TS0021  TS0022  TS0023  TS0069
```

Command	Description
DISK	Selects online disks.

Type:
Major

An argument of AL (DISKAL) displays all offline disks as well as those online.

Command	Description
DPAV	Displays PAV (parallel access volume) base devices.

Type:
Major

Command	Description
DSKB	Selects busy disks.

Type:
Major

Command	Description
DSKC	Selects disks with suspended channel programs.

Type:
Major

Command	Description
DSKE	Selects permanently resident disks.

Type:
Major

Command	Description
DSKG	Selects mass storage (MSS) virtual disks.

Type:
Major

Command	Description
DSKM	Selects disks waiting on mounts.

Type:
Major

Command	Description
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
```

Command	Description
DSKP	Selects DASD volumes with a mount status of PUBLIC.

Type:
Major

Command	Description
DSKQ	Selects disks with I/O queue length of <i>nn</i> or more.

Type:
Major

Format:
DSKQnn

The `DPLTnn` minor command plots a microscopic analysis of device utilization. You can use it to investigate disks with I/O queues.

Command	Description
DSKR	Selects disks with a RESERVE currently issued from this CPU.

Type:
Major

Command	Description
DSKS	Selects DASD volumes with a mount status of STORAGE.

Type:
Major

Command	Description
DSKU	Selects disks with UCBnames starting with <i>xx</i> .

Type:
Major

Format:
DSKUxx

The variable *xx* specifies the UCBnames with which the disks begin. This may or may not correspond to control unit *xx*. `DSKUxx` does not consider alternate paths.

The following example selects disks with UCBnames that start with 58, which includes disks 580 through 58F:

```
DSKU58
```

Command	Description
DSKV	Selects DASD volumes with a mount status of PRIVATE.

Type:
Major

Command	Description
GDEV	Lists devices with name ccccc.

Type:

Major

Format:

nGDEV ccccc

This major command lists disk and tape devices with the generic name ccccc. The value of *n* can be one of the following: the less-than symbol (<) to display all devices, or the numbers 1 to 9 and the letters A to Z (representing 10 to 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:

```
<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.

Command	Description
3330	Displays 3330 disks.

Type:

Major

Format:

3330nn

An argument of 11 displays 3330-11 disks.

Command	Description
3340	Displays 3340 disks.

Type:

Major

Command	Description
3350	Displays 3350 disks.

Type:

Major

Format

3350c

An argument of P displays the disks attached to 3880-11 or 3880-21 buffered-paging facility control units. In addition, for non-base exposures, the unit address and exposure number appear instead of the volser.

Command	Description
3375	Displays 3375 disks.

Type:

Major

Command	Description
3380	Displays 3380 disks.

Type:
Major

Command	Description
3390	Displays 3390 disks.

Type:
Major

Disk space immediate command

Command	Description
SVOL cccccc	Displays available space on disk with volser=ccccc.

Type:
Immediate

For Extended Address Volumes (EAVs), supported on z/OS V1.10 and later, SVOL displays both total volume and track-managed information. For non-EAVs, the total and track-managed data is identical.

```
SVOL CBEAV3
>SVOL CBEAV3 8402 PRV/RSDNT EAV
>TOT VOLUME FREE(000240545,0000000015) AREAS=0000000004 MAX_CNTG(000175056,00)
>TRK-MNGD FREE(000065300,0000000015) AREAS=0000000002 MAX_CNTG(000065300,10)
```

Figure 56. Output of SVOL command for EAV

where:

8402

Is the 4-digit device number.

PRV/RSDNT

Is the allocation (PRV = Private) and mount status (RSDNT = Permanently Resident) of the device.

EAV

Indicates that this is an Extended Address Volume. (Blank if non-EAV.)

TOT VOLUME and TRK-MNGD (EAV)

- FREE=(*nnnn*, where *mmmm*)*nnnn* is the total amount of free cylinders and *mmmm* is the amount of additional free tracks.
- AREAS=*nnnn*, where *nnnn* is the number of areas that subdivide (or fragment) the available free space.
- MAX_CNTG=(*nnnn*, *mmmm*). The largest amount of contiguous free space available on this device. *nnnn* is the number of cylinders in the largest free extent and *mmmm* is the number of additional tracks in the largest free extent.

To reduce overhead, this command places a comment symbol (>) in column 1 after it executes. To prevent the self-comment function, place a less-than symbol (<) in column 1.

Selected disk minor commands

You can use the following minor commands immediately after a disk major command to display information about the disk drives selected by the disk major commands.

Command	Description
DADR	Displays the unit address of the device.

Type:

Minor of disk majors

Command	Description
DALC	Displays number of allocations to the device.

Type:

Minor of disk majors

Command	Description
DCAT	Displays whether a device is static, installation-static, or dynamic.

Type:

Minor of device majors, for example, DISK.

DCAT will display STATIC, I-STATIC, or DYNAMIC, to indicate the device category.

Command	Description
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.

Command	Description
DIOQ	Displays I/O queue length on the disk.

Type:

Minor of disk majors

Command	Description
DOPN	Displays number of open DCBs and ACBs on the device.

Type:

Minor of disk majors

Command	Description
DPIN	Displays whether a device is pinned.

Type:

Minor of device majors, for example, DISK.

DPIN will display PINNED, UNKNOWN, or a field of blanks, to indicate the device category.

Command	Description
DPLT	Displays device activity every <i>nn</i> milliseconds.

Cyl

Cylinder address. (The > indicates the point at which a new cylinder was accessed.) Rls in this field indicates that a standalone release is in progress, and therefore no cylinder is involved.

Nrdy

Not ready.

Resv

Device reserved this CPU.

The above example shows that the DSKQ01 command selected a disk with a volser of MVS005 because it had an I/O queue length of one or more. The minor command DPLT03 plots the activity on the selected device at 3 millisecond intervals (OMEGAMON 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 we specified a 3 millisecond interval (DPLT03), the actual interval that appears is 4 milliseconds (Interval: 4). This is because other address spaces operated at a higher priority than OMEGAMON at the time.

Command	Description
DRES	Displays device reserve count from this CPU.

Type:

Minor of disk majors

Command	Description
DSTA	Displays mount status.

Type:

Minor of disk majors

Status can be:

- PRIVATE
- PUBLIC
- STORAGE

Command	Description
DTYP	Displays disk type.

Type:

Minor of disk majors

Disk type can be 3380, 3390, etc.

Command	Description
DUSR	Displays current user of device.

Type:

Minor of disk majors

Command	Description
DVMP	Displays unit control block (UCB) hex dump.

Type:

Minor of disk majors

DVMP 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

```

Command	Description
DVOL	Displays the volser of a selected device.

Type:

Minor of DEV and DEVL

Command	Description
ICHPn	Displays installed channel paths for the disk.

Type:

Minor of disk majors

The value of *n* is a number from 1 to 8.

Command	Description
OCHP	Displays online channel paths for the disk.

Type:

Minor of disk majors

Format:

OCHPn

The value of *n* is a number from 1 to 8.

Examples of disk commands

The following are examples of how to use the disk commands.

This example shows 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      ....

```

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  MVSRES  SPOOL
dusr  *MASTER*  JES2
dioq   3        2
dadr   163     167

```

The next example selects all private disks.

```

DSKV   DB0021  DB0022  DB0023  DB0024  KN0056  KB0087
dusr  MESSAGE1
dio  .R      2.3          MESSAGE1  5.3

```

The next example selects all disks whose volsers begin with WO.

```

DSKNWO  WORK01  WORK02
dadr    140    147

```

Disk I/O information

Command	Description
XDEV	Displays device I/O information.

Type:

Immediate

When the I/O Supervisor (IOS) tries to start an I/O operation against a device, z/OS queues the I/O on the device's Unit Control Block (UCB), whether the I/O hardware actually accepted the I/O or not. Therefore, inspection of the contents of the device queues indicates what I/O is currently active and what I/O did not actually start. XDEV displays the contents of the I/O queue for all devices, with each I/O request on a different line.

An R in the label field excludes I/O for 3850 MSS virtual storage volumes. An X in the label field excludes suspended channel programs from the display. Suspended channel programs are frequently outstanding against page volumes. Since these are harmless and usually clutter the display, you can suppress them.

When you use XDEV, keep the time dimension in perspective. The screen may update every 5 seconds. However, since the I/O rate on devices can be 30 I/Os per second and 100 I/Os per second on the channel, under some conditions the OMEGAMON user can miss 150 I/Os per device, or 500 I/Os per channel every 5 seconds.

Remember that the device queue display is just a snapshot. If there are a number of outstanding I/Os against a device for a continuing period of time, a problem may result. If the order of the waiting I/Os does not change after several cycles, there is probably an I/O lockout condition. Example:

```

XDEV  Unit  volser  Jobname  I/O  Drvr  Cyl  Device Status
+    161  MVS002  PAYROLL  SSCH  EXCP  280
+    180  PAGE01  *MASTER*  SSCH  EXCP  24  Channel Pgm Suspended
+    204  SYS001  TSOA13   SSCH  Ftch  93

```

XDEV provides the following information:

Unit

Device address

volser

Device volser

Jobname

Jobname of the I/O requester

I/O

I/O type or Wait (usually SSCH Start Subchannel)

Drv

Type of I/O-driver (EXCP, JES3, VSAM, VTAM...)

Cyl

Target decimal cylinder address (DASD only)

Device Status

Unusual device status information (such as not ready or mount pending)

The example shows:

- PAYROLL and SORT competing for the same device (MVS002)
- A suspended channel program against the PAGE01 page volume
- A TSO user with a currently active fetch against SYS001

Tape information

This section describes major commands for tape drives.

Tape select major commands

Major commands to select tape drives provide a number of different ways to examine the tape configuration. For each tape drive selected by a major command, a number of minor commands can describe the device in further detail.

Command	Description
TAPE	Selects all online tape drives.

Type:

Major

An argument of AL (TAPEAL) displays all offline tape drives as well as those online.

Command	Description
TPFR	Selects online and free (unallocated) tape drives.

Type:

Major

Command	Description
TPMT	Selects tape drives with mounts pending.

Type:

Major

Command	Description
TPOF	Selects all offline tape drives.

Type:

Major

Command	Description
TP16	Selects 1600 BPI tape drives.

Type:
Major

Command	Description
TP38	Selects 3480 tape drives.

Type:
Major

Command	Description
TP62	Selects 6250 BPI tape drives.

Type:
Major

Command	Description
TP7T	Selects 7-track tape drives.

Type:
Major

Command	Description
TP80	Selects 800 BPI tape drives.

Type:
Major

Long-term device utilization

Long-term device utilization displays are available if RMF is active for the desired devices. When OMEGAMON first starts up, these numbers represent averages since OMEGAMON initialization. The first time an RMF interval begins, OMEGAMON synchronizes with it.

Three dimensions are at work when you display long-term data:

- Device group to appear
- Selection thresholds
- Display format

Long-term device utilization major command

Command	Description
STAT	Displays statistical data collected from RMF.

Type:
Major

STAT displays the following RMF statistics:

- interval start time
- interval elapsed time
- length of interval

Minor commands that you enter beneath the STAT major command display device service time statistics in numbered format.

An argument of P with STAT (STATP) plots device service time statistics in graph format, where each point represents 10%.

An argument of Z with either STAT or STATP sets all the thresholds of the minor commands that do not appear on the screen to zero. Thus, you do not need to re-enter every STAT minor command in order to set their thresholds to 0; just enter STATZ or STATPZ. Then, you can enter only those minor commands for which you want to change the threshold.

Example:

```
STAT Interval Start Time: 11:45:01 Elapsed: 43 SEC Length: 15:00 MN
```

“[Long-term device utilization minor commands](#)” on page 199 explains minor commands.

Long-term device utilization minor commands

The names of the following minor commands of STAT all have the following form.

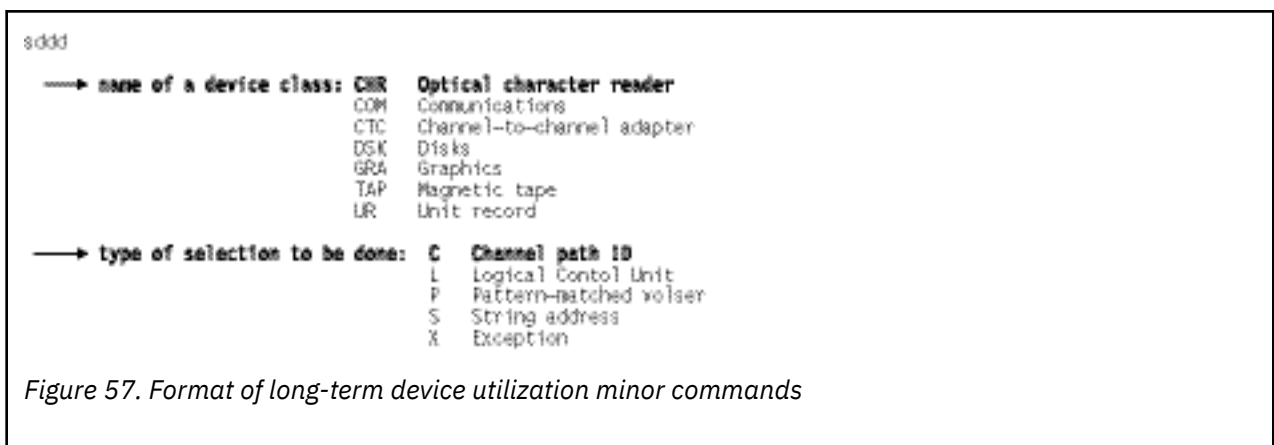


Figure 57. Format of long-term device utilization minor commands

Not all combinations are possible. For example, while there is a PDSK minor command to display all the DASD devices whose volsers match a certain pattern, there is no such thing as a PCTC command, because channel-to-channel adapters don't have volsers.

See [Figure 58 on page 199](#) for an example of the display format of all these commands.

These commands display information about long-term device utilization.

```
STAT Interval Start Time: 10:00:00 Elapsed: 7:17 MN Length: 14:59 MN
pdsk * <-- Volser Pattern
+ Number of Devices = 1389 Number of Samples = 438
+ Volser LCU Rate Util% PAV Resp = IOSQ + Pend + Conn + Disc RESV% #Alloc
+3220 PP010P 046 .3 .2 .1 40.0
+3221 PP011P 046 .5 .3 .2
+3222 PP012P 046 .7 .4 .3 51.0
+3223 PP013P 046 .7 .4 .3
+3224 PP014P 046 .5 .2 .2 5.0
+3225 PP020P 046 1.0 .5 .4
+3226 PP030P 046 .1 .4 .2 .2 4.0
+3227 PP040P 046 .6 .4 .2 1.0
+3228 PP050P 046 2.7 .1 .8 .2 .5 12.0
+3229 PP060P 046 .6 .4 .2
+322A PP061P 046 .4 .3 .1
+322B PP062P 046 .4 .3 .1
+322C PP070P 046 2.1 .2 1.3 .2 1.1 74.0
```

Figure 58. Typical minor command of STAT display

volser

Device volume serial number

LCU

Logical Control Unit

Rate

I/O rate (I/Os per second)

Util%

Device utilization in percent

PAV

Number of parallel access volumes

Resp

Average total device response time in milliseconds

IOSQ

Average I/O supervisor queuing time in milliseconds

Pend

Average pending time in milliseconds

Conn

Average connect time in milliseconds

Disc

Average disconnect time in milliseconds

RESV%

Percent of the time RESERVED from this CPU

#Open

Average number of open data sets

An argument of P with STAT (STATP) displays the statistics in a graph:

```

STATP Interval Start Time: 15:30:04 Elapsed: 8:04 MN Length: 14:55 MN
SDSK35 Number of Devices = 208 Number of Samples = 485
+ volser LCU UTIL% RESV%
+ 350 DB2003 00A
+ 351 TS0023 00A>
+ 352 PROD06 00A
+ 353 SYSA25 00A-->
+ 354 PROD02 00A
+ 355 TS0025 00A
+ 356 OMON29 00A
+ 357 MP220B 00A
+ 358 PROD07 00A
+ 359 COM007 00A
+ 35A QM0002 00A
+ 35B COM008 00A

```

Figure 59. Typical minor command of STATP display

This data pertains to the current RMF interval; the STAT command displays how long the intervals are, and how much of the current interval elapsed.

Commands that set performance thresholds

These minor commands select any device of the indicated type that was assigned to Logical Control Unit (LCU) xxx, where xxx is a hexadecimal LCU ID. The system assigns these LCU numbers during the IOCDs generation.

Command	Description
DSCnnn	Sets threshold for average device disconnect time > nnn milliseconds.

Type:

Minor of STAT

Command	Description
DUTnnn	Sets threshold for average device utilization > <i>nnn</i> percent.

Type:
Minor of STAT

Command	Description
IOSnnn	Sets threshold for average IOS queue time > <i>nnn</i> milliseconds per I/O.

Type:
Minor of STAT

Command	Description
PNDnnn	Sets threshold for average device pending time > <i>nnn</i> milliseconds.

Type:
Minor of STAT

Command	Description
RSPnnn	Sets threshold for average total device response time > <i>nnn</i> milliseconds.

Type:
Minor of STAT

Commands that display utilization by channel path ID

The following commands display long-term device utilization by Channel Path ID and Device Type:

Command	Description
CCHRxx	Displays optical character readers connected by channel path xx.

Type:
Minor of STAT

Command	Description
CCOMxx	Displays communications devices connected by channel path xx.

Type:
Minor of STAT

Command	Description
CCTCxx	Displays channel-to-channel adapters connected by channel path xx.

Type:
Minor of STAT

Command	Description
CDSKxx	Displays disks connected by channel path xx.

Type:
Minor of STAT

Command	Description
CGRAXx	Displays graphics devices connected by channel path xx.

Type:
Minor of STAT

Command	Description
CONnnn	Displays and sets threshold for average device connect time > nnn milliseconds.

Type:
Minor of STAT

Command	Description
CTAPxx	Displays magnetic tapes connected by channel path xx.

Type:
Minor of STAT

Command	Description
CUR xx	Displays unit record devices connected by channel path xx.

Type:
Minor of STAT

Commands that display utilization by logical control unit

The commands below display long-term device utilization by logical control unit (LCU) and device type.

Command	Description
LCHRxxx	Displays optical character readers connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LCOMxxx	Displays communications devices connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LCTCxxx	Displays channel-to-channel adapters connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LDSKxxx	Displays disks connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LGRAxxx	Displays graphics devices connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LTAPxxx	Displays magnetic tapes connected by LCU xxx.

Type:
Minor of STAT

Command	Description
LUR xxx	Displays unit record devices connected by LCU xxx.

Type:
Minor of STAT

Commands that display utilization by volser pattern

The commands below display long-term device utilization by pattern-matched volser. These minor commands display any device of the indicated type whose volser matches the pattern you specify in the string *aaaaaa*. A pattern consists of alphanumeric characters and asterisks (*), which are wildcard characters. For example, the pattern *MVS00** displays the devices *MVS001*, *MVS002*, *MVS003*, and so on.

If an asterisk is the last character, in the pattern any character after that point matches. For example, the pattern *VS** displays the volumes *VSRESA*, *VSAM01*, *VS*, and so on.

Command	Description
EDSK cccccc	Displays disk information by volser pattern cccccc.

Type:
Minor of STAT

This display supplements the PDSK display and includes control unit busy, device busy, and ESCON switch port busy delay times.

Command	Description
PDSK cccccc	Displays disks by volser pattern cccccc.

Type:
Minor of STAT

Command	Description
PTAP cccccc	Displays magnetic tape devices by volser pattern cccccc.

Type:
Minor of STAT

Commands that display utilization by string address

The following commands display long-term device utilization by string address:

Command	Description
SCHRxx	Displays optical character readers belonging to string xx.

Type:

Minor of STAT

Command	Description
SCOMxx	Displays communications devices belonging to string xx.

Type:

Minor of STAT

Command	Description
SCTCxx	Displays channel-to-channel adapters belonging to string xx.

Type:

Minor of STAT

Command	Description
SDSKxx	Displays disks belonging to string xx.

Type:

Minor of STAT

Command	Description
SGRAXx	Displays graphics devices belonging to string xx.

Type:

Minor of STAT

Command	Description
STAPxx	Displays magnetic tape devices belonging to string xx.

Type:

Minor of STAT

Command	Description
SUR xx	Displays unit record devices belonging to string xx.

Type:

Minor of STAT

Commands that display long-term device utilization by exception

The following commands display long-term device utilization by exception:

Command	Description
XDSK	Displays DASD devices which exceed certain thresholds.

Type:

Minor of STAT

XDSK displays devices that exceed the thresholds described in the following minor commands.

Long-term channel path utilization

Command	Description
CHNP	Displays long-term channel information.

Type:

Minor of STAT

CHNP is a minor command of STAT, which shows channel utilization since the last RMF interval. When OMEGAMON first starts up, these numbers represent averages since OMEGAMON initialization. The first time an RMF interval starts up, OMEGAMON automatically synchronizes with it.

The following figure shows an example of a long-term channel utilization display.

```

STAT      Interval Start Time: 10:00:01      Elapsed:  34:13 MN      Length:   1 HR
CHNP      Number of Samples = 245
+  ID  Type  % Busy      ID  Type  % Busy      ID  Type  % Busy
+  00  Blk   13.2      10  Blk   24.2      20  Byte
+  01  Blk                                     21  Byte   16.9
+  02  Byte                                     22  Byte
+  03  Byte    6.5      13  Byte   53.2      23  Byte
+  04  Byte                                     24  Byte   23.1
+  05  Byte   23.2      15  Byte
+  06  Byte    8.9      16  Byte
+  07  Byte                                     27  Byte
    
```

Number of Samples

This is determined by SRM, not by the RMF.

ID

Channel Path ID number.

Type

Channel type:

Blk

Block Multiplexor

Byte

Byte Multiplexor

ES-CHN

ESCON channel

ES-CNV

ESCON converter

ES-DIR

ESCON director

% Busy

Channel Path utilization in percent

RMF must be monitoring channels to show long-term utilization.

Long-term logical control unit (LCU) utilization

Command	Description
LCU xxx xxx	Selects and supplies information about all devices attached to LCUs xxx through xxx.

Type:

Minor of STAT

The variable xxx is a hexadecimal LCU ID. The system assigns LCU numbers during the IOCDS generation. The minimum value for the first LCU in the range is 0, and the maximum value for the second one in the range is FFF.

If you enter the command with only one LCU number (that is, LCU xxx), data appears only for devices on the single LCU.

If you enter the command without an argument, devices attached to all LCUs appear.

For example:

```

STAT Interval Start Time: 13:59:59      Elapsed: 34:14 MN      Length: 1 HR
LCU 000 FFF >> Number of Samples = 187 <<
+ ID Contention Delay Q % All CHP CHP Control CHP % CU
+ -- ---Rate--- - Length ---Busy-- --- -Units- Taken Busy
+ 009 1.164 .14 1.728 16 161 26.021 8.870

```

Figure 60. Typical LCU display

ID

Logical Control Unit Number.

Contention Rate

Rate per second at which the I/O processor places delayed I/O requests on the CU-HDR for this LCU.

Delay Q Length

Average number of requests on the LCU Control Word (CUCW).

% All CHP Busy

Percentage of time during the interval when all channel paths belonging to the LCU were busy at the same time.

CHP

Channel Path ID number of this LCU.

CONTROL UNITS

Control Units comprising this LCU.

CHP Taken

Rate at which I/O requests to devices of this LCU are satisfied by each CHPID during the interval.

% CU Busy

Ratio of the number of requests deferred because the control unit was busy and the number of attempts to service I/O requests during the measurement interval.

Chapter 10. Address space information

In the z/OS operating system, many programs can execute at the same time. Some of these programs are batch jobs; others are TSO sessions; still others are started tasks such as JES2/3, TCAM, VTAM, CICS, and IMS. In the language of z/OS, all of these executing programs are referred to as address spaces.

Address space information major commands

The address space major commands select specific address spaces, such as those that are resident, CPU-dispatchable, and batch jobs.

The .SPT immediate command lets you set patterns for the selection of address spaces. See the description of .SPT in [Chapter 3, “Operational commands,”](#) on page 35 for details.

ALLJ

Selects all address spaces in ASID order.

Type:

Major

Address spaces may be further broken down into three types:

Batch jobs

BATJ command

Started tasks

STCJ command

TSO users

TSOJ command

In addition to all active address spaces, ALLJ shows idle initiators as INIT.

BATJ

Selects all initiators, active or not.

Type:

Major

BATJ shows idle initiators as INIT.

BATL

Selects batch jobs in logically swapped status.

Type:

Major

BATX

Selects active batch jobs.

Type:

Major

BCPU

Selects CPU dispatchable batch jobs.

Type:

Major

BIO

Selects non-CPU dispatchable batch doing I/O.

Type:

Major

These are batch jobs that have done I/O in the last five seconds and are not CPU-dispatchable now.

BSWP

Selects batch jobs physically swapped out.

Type:

Major

These are batch jobs that are waiting and are physically swapped out.

BWAT

Selects batch waiting (resident).

Type:

Major

These are batch jobs that are currently waiting and are in main storage. These jobs often become non-swappable because they are waiting on tape mounts.

JAS#nn

Selects address space with ASID=#nn.

Type:

Major

This value is in decimal.

This command selects one address space only; therefore use it to monitor specific address spaces.

JAS# issues a message if the selected address space is unusable.

If an ASID has three digits, use JASnnn.

JCPU

Selects CPU dispatchable address spaces.

Type:

Major

JLSW

Selects address spaces in logically swapped status.

Type:

Major

These are address spaces that are waiting and are logically swapped out.

JOBN ccccccc

Selects address space ccccccc.

Type:

Major

In dedicated mode, you must position the cursor on another line before the command takes effect.

JPAG

Selects all address spaces waiting on a page operation.

Type:

Major

JRES

Selects resident address spaces.

Type:

Major

JRES selects:

- non-swappable system tasks (such as *MASTER*, JES2, TCAM, VTAM, and CICS)
- executing batch jobs
- batch jobs waiting on tape mounts only

- executing TSO users

JSNAnn

Selects address spaces with dispatching number *nn*.

Type:

Major

JSNAnn displays as many address spaces as can fit on a line, by their cpu dispatching sequence order beginning with number *nn*.

JSNCnn

Selects CPU dispatchable address space with dispatching number *nn*.

Type:

Major

JSNCnn is the same as JSNAnn, except that it shows only address spaces ready to execute (contending for CPU cycles).

JSWP

Selects physically swapped address spaces.

Type:

Major

JSWP selects idle address spaces that are swappable and swapped out, such as:

- RMF
- batch jobs waiting (except those waiting on tape mounts)
- jobs swapped by SRM
- TSO users not in transaction
- idle initiators

STCJ

Selects started task.

Type:

Major

These address spaces include system tasks, such as the MASTER scheduler, JES2, TCAM/VTAM, and any other address spaces for which a START command has been issued.

STCL

Selects started tasks in logically swapped status.

Type:

Major

TSOJ

Selects TSO users logged on.

Type:

Major

The user does not have to be active.

Sometimes the word *LOGON* appears. This means that a TSO address space was created, but the logon process did not complete. This is analogous to an idle initiator for batch address spaces.

TSOL

Selects TSO users in logically swapped status.

Type:

Major

TSOX

Selects TSO users in a transaction.

Type:

Major

This refers to users who pressed Enter and are waiting for the command to complete.

Address space information minor commands

After you choose a major command to select the type of address space, you can choose minor commands to display more detailed information about the address space.

The address space minor commands are grouped into three categories:

Identification

Includes ASID, proc, or step

Environment

TCB time, SRB time, I/O rate

Storage activity

Real and virtual storage

CPU activity

SRB and TCB

SRM activity

Paging, service units, swapping

Time information

Time in transaction, in current period, since last terminal wait

Miscellaneous

Job status, I/O count, short wait count

Address space identification

ASID

Displays address space ID in hex (in parentheses) and decimal.

Type:

Minor of address space majors

Format

[X]ASID where X can be:

blank

Displays both hexadecimal and decimal ASID.

D

Displays only decimal ASID.

H

Displays only hexadecimal ASID.

Address spaces are numbered beginning with the Master Scheduler at ASID 01. As z/OS starts address spaces, they are assigned an ASID number, which z/OS uses as an internal reference.

.DMPxx

Dumps 4 hex bytes at offset xx from ASCB or User Control Block (UCB).

Type:

Minor of address space majors

This command is considered a generalized minor command because it applies to more than one group of major commands.

When you use .DMPxx with an address space major command, it displays four hex bytes at offset xx from the ASCB or the UCB.

When you use .DMPxx with a device major command, it displays four hex bytes at offset xx from the device's UCB.

If you place the letter C in column 1, eight characters at offset xx are dumped in character format.

If you want to specify a three-digit offset, use the .DMxxx command.

PROC

Displays procedure stepname.

Type:

Minor of address space majors

PROC displays **INIT** when a batch job is moving between steps. This tells you when the CPU time figures pertain to the initiator rather than to your job.

STEP

Displays stepname.

Type:

Minor of address space majors

STEP displays **INIT** when a batch job is moving between steps. This tells you when the CPU time figures pertain to the initiator rather than to your job.

Address space environment

AENV

Displays graphic display for a given user.

Type:

Minor of address space majors

AENV displays graphic environmental data on the address space over both the last interval and the last 20 intervals. For example:

```

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(%) 1.3 |> . . . . .| 1.3 |> . . . . .|+
+ SRB Time(%) .0 |> . . . . .| .0 |> . . . . .|
+ AST Time(%) .0 |> . . . . .| .0 |> . . . . .|
+ IFA Time(%) n/a |> . . . . .| n/a |> . . . . .|
+ IFC Time(%) .0 |> . . . . .| .0 |> . . . . .|
+ IIP Time(%) n/a |> . . . . .| n/a |> . . . . .|
+ IIC Time(%) .0 |> . . . . .| .0 |> . . . . .|
+ Page-ins/s_ .0 |> . . . . .| .0 |> . . . . .|
+ Page-outs/s .0 |> . . . . .| .0 |> . . . . .|
+ I/O's/sec__ .0 |> . . . . .| .0 |> . . . . .|
+ SU's/sec___ 3137 |-----> . . . . .| 3137 |-----> . . . . .|+
+ Working Set28840K |+++++++| 28840K |+++++++|+
+ Pg-in/CPU-s .0 |> . . . . .| .0 |> . . . . .|

```

The page-in and page-out rates are in seconds. Pg-in/CPU-s are the number of page-ins per second while the task has control of the CPU.

The middle (short-term) portion of the display represents the last OMEGAMON cycle; the right (long-term) portion represents a running average of the previous 20 OMEGAMON cycles. In the last column of the long-term display, a + indicates the trend is upward, a - indicates the trend is downward.

TRACnn

Displays plot address space activity at nn millisecond intervals.

Type:

Minor of address space majors

where nn specifies the interval between samples (01 - 99 milliseconds).

OMEGAMON exception analysis automatically warns when an address space waited longer than a user-specified threshold. Often, however, address spaces suffer from degradation caused by

interaction with other address spaces. Diagnosing such problems is difficult. The TRAC command helps explore the reasons for address space degradation by closely analyzing many factors associated with the address space. It tracks the activity of an address space.

TRAC samples fifty times between screen refreshes.

The interval value on the right is the true interval between samples, calculated after TRAC completes processing. On a system running perfectly, this number is the same as the *nn* sampling interval. This number may vary due to your system's workload.

As the plot progresses from left to right, a number of fields reflect either the status of the address space or its activity since the last sample. Each sample is represented by one column in the display.

When a number appears for a particular sample, A-Z represent the numbers 10-35. Beyond Z, a plus sign (+) appears.

TRAC is very effective with the JPAG major command, which selects any address space currently waiting on a paging operation.

In the following figure, BATX selects batch jobs and TRAC01 tracks the first batch job YCAN006, with a sampling time of one millisecond. Although a one millisecond sample was requested, the result was an average of three milliseconds for each of the fifty samples. The reason for this is that higher priority work in the system prevented OMEGAMON from getting control every millisecond. This difference is usually not as great in dedicated mode with OMEGAMON running at a high priority.

```

BATX  YCAN006  YCAN005  ASYD
trac10 ----- OMEGAMON Peek at: K3DSST22 -----
+      CPU
+      SRBT  S
+      SRBP
+      SRBC
+      IFA
+      IFC
+      IIP
+      IIC
+      IOJ
+      SWCT
+      Swap
+      Disp .....
+      Seqn
+      Pgin
+      V-in
+      CAin
+      PgwT
+      Dev
+      Stat
+      Cyl
+      User
+      Rcyl
Samples: 50
Interval: 10

```

CPU

T indicates address space used TCB CPU time since last sample.

SRBT

S indicates the address space used non-preemptable SRB CPU time since last sample.

SRBP

SRBC

IFA

Series z Application Assist Processor (zAAP) consumed during the time-slice.

IFC

Standard processor consumed during time-slice performing zAAP-eligible work

IIP

System z Integrated Information Processor (zIIP) consumed during time-slice.

IIC

Standard CP consumed during time-slice performing zIIP-eligible work

IOJ minor

Indicates the number of I/Os issued since last sample.

SWCT

Indicates the number of short waits since last sample.

SWAP

S indicates address space is swapped out.

DISP

Indicates user is waiting on resource as follows:

- Address space is waiting
- | Address space is CPU-dispatchable

SEQN

Obsolete

PGIN

Indicates the number of non-common area, demand page-faults since last sample.

V-IN

Indicates number of VIO page-faults since last sample.

CAIN

Indicates number of common area page-faults since last sample.

PGWT

P indicates address space is currently waiting on a page operation.

DEV

Indicates address space is waiting on a device at address *cuu*. (> indicates the point at which a new device is accessed.)

STAT

Indicates device status:

- D** Device busy
- C** Control unit busy
- +** Device + control unit busy (possible RESERVE on shared DASD)
- H** Waiting on channel
- N** Device not ready
- M** Device mount pending
- S** A channel program is currently active in this subchannel
- ?** One of the following:
 - There is a suspended channel program.
 - There is an IOSB queued to the UCB.
 - The unit is in undetermined status.

CYL

Indicates cylinder address of pending I/O (for disks). (> indicates the point at which a new cylinder is accessed.)

USER

Indicates current user of device (disk only). (> indicates the point at which a new user accesses the device.)

RCYL

Indicates cylinder address of current user. (> indicates the point at which a new cylinder is accessed.)

From this display we can see that YCAN006 waited on I/Os most of the time; the DISP line is mostly dots. Program cycles as indicated by the T (TCB time) of CPU were used to initiate new I/Os. I/Os (as noted by the 1 count on IOJ) are always associated with SRB activity (see S on SRBT line). The address space performed no page-ins and did not wait on page operations. The address space wait activity was associated with I/Os specifically on device address 14E. The I/O wait activity on 14E was caused by a device busy status (STAT), which resulted from competition with address space YCAN005, as well as from competition of YCAN006 with itself at cylinder addresses 142 and 239.

Address space storage activity

CSA

Displays a job's usage of CSA (requires the CSA Analyzer).

Type:

Minor of address space majors

DSPC

Displays dataspace utilization (in pages) by an address space.

Type:

Minor of address space majors

DSPC displays the number of 4K blocks of virtual storage in use. Only user key data spaces are included.

ECSA

Displays a job's usage of ECSA (requires the CSA Analyzer).

Type:

Minor of address space majors

ESQA

Displays a job's usage of ESQA (requires the CSA Analyzer).

Type:

Minor of address space majors

FIXF

Displays number of fixed frames for swapped-out user.

Type:

Minor of address space majors

The term IN displays if the user is swapped in.

FMCT

Displays total frame count of a swapped in address space.

Type:

Minor of address space majors

Each frame is 4096 bytes.

The working set size is calculated as $4 * FMCT$. You can view this with the WKST minor command.

For FMCT and WKST, the value is zero when the address space is physically swapped out. Instead of displaying the zero value, the last working set size (or last frame count) appears and is indicated with an S. When the allocated frame count is 0, address spaces may be resident and still show an S for FMCT or WKST.

FXFB

Displays the number of fixed frames below 16M for an address space.

FXFB displays a number that represents the real storage frames below the 16M line that are fixed and cannot be moved to disk to make room for other address spaces. If the address space is swapped out, FXFB displays zero.

FXFR

Displays number of fixed frames in storage.

Type:

Minor of address space majors

These are the number of real storage frames an address space is using that are fixed and cannot be moved to disk (as slots) to make room for other address spaces.

The term Swapped displays if the address space is physically swapped out.

The FXFR exception produces a warning for any address space whose fixed frame count exceeds a threshold.

NVSC

Displays non-VIO slots held.

Type:

Minor of address space majors

These are the slots on a page data set used by portions of the program.

This count includes the number of frames swapped out if the address space is swapped out.

SQA

Displays a job's usage of SQA (requires the CSA Analyzer).

Type:

Minor of address space majors

TWSF

Displays target working set size in frames.

Type:

Minor of address space majors

TWSS

Displays target working set size.

Type:

Minor of address space majors

Display size in K (1024 bytes).

VSC

Displays VIO slots.

Type:

Minor of address space majors

This is the number of VIO slots used by an address space on page data sets.

WKST

Displays working set size in K.

Type:

Minor of address space majors

WKST is calculated as $4 * \text{the frame count of a swapped in address space (FMCT)}$.

For FMCT and WKST, the value is zero when the address space is physically swapped out. Instead of displaying the zero value, the last working set size (or last frame count) appears and is indicated with

an S. When the allocated frame count is 0, address spaces may be resident and still show an S for FMCT or WKST.

WSSI

Displays working set size at last swap in.

Type:

Minor of address space majors

Displays frame count.

Address space CPU activity

CPGS

Displays CPU page-seconds.

Type:

Minor of address space majors

CPU

Displays TCB time for current job step (batch) or user session (TSO).

Type:

Minor of address space majors

The value shown is in whole seconds.

CPU2

Displays TCB time in seconds.

Type:

Minor of address space majors

The value shown is in seconds plus fractions of seconds.

An argument of .R gives the percentage of the total time (of an interval) attributable to TCB utilization.

The system resets TCB time to zero when a job step completes, so .R may display an incorrect value if the TCB time is reset between OMEGAMON cycles.

CPUL

Displays job step CPU time limit.

Type:

Minor of address space majors

When job step timing is turned off by specifying TIME=1440 or TIME=NOLIMIT, No Limit is displayed.

DISP

Displays dispatching algorithm in use for a job.

Type:

Minor of address space majors

The algorithm can be one of the following:

- MTW (Mean-Time-to-Wait)
- ROTATE
- TIME-SLC
- FIXED

DPRT

Displays dispatching priority.

Type:

Minor of address space majors

The priority appears in hex and decimal.

DVCT

Displays device connect time in seconds.

Type:

Minor of address space majors

ECPU

Displays the independent enclave time consumed by the address space in seconds. ECPU displays the equivalent of the SMF Type 30 field SMF30ENC.

Type:

Minor of address space majors

Format:

ECPU

Use with a .R suffix to display independent enclave time as a rate in milliseconds per second; use with .% suffix to display the independent enclave time as a percentage to one decimal place (tenths of a percent).

IFAC

Displays time spent by the address space executing work eligible for a Series z Application Assist processor (IFA) on a standard processor.

Type:

Minor of address space majors

IFAT

Displays time spent by the address space executing on a Series z Application Assist (IFA) processor.

Type:

Minor of address space majors

IFAX

Displays time spent by the address space executing work on a standard processor.

Type:

Minor of address space majors

IIPC

Displays time spent by the address space executing work eligible for a System z Integrated Information Processors (zIIP) on a standard processor.

Type:

Minor of address space majors

IIFE

Displays time spent by enclaves in this address space executing System z Integrated Information Processors (zIIP) work.

Type:

Minor of address space majors

IIPP

Displays time enclaves owned by this address space executed on a System z Integrated Information Processor (zIIP).

Type:

Minor of address space majors

IIPS

Displays time spent by the address space executing in SRB-mode on a standard processor.

Type:

Minor of address space majors

IIPT

Displays time spent by the address space executing on a System z9 Integrated Information Processor (zIIP).

Type:

Minor of address space majors

JCAF

Displays CPU affinity.

Type:

Minor of address space majors

RCP%

Displays the percentage of step CPU time used.

Type:

Minor of address space majors

When job step timing is turned off by specifying TIME=1440 or TIME=NOLIMIT, N/A is displayed.

RCPU

Displays the CPU time remaining for the job step.

Type:

Minor of address space majors

When job step timing is turned off by specifying TIME=1440 or TIME=NOLIMIT, No Limit is displayed.

SRBT

Displays SRB time in whole seconds.

Type:

Minor of address space majors

This is for the current job step (batch) or user session (TSO).

SRB2

Displays SRB time in seconds with a decimal fractional part.

Type:

Minor of address space majors

TCP2

Displays total CPU utilization in seconds with a decimal fractional part.

Type:

Minor of address space majors

This is calculated as:

```
TCB + SRB time
```

Address space SRM activity

JPCI

Displays address space common area page-ins.

Type:

Minor of address space majors

To display this value as a rate, use an argument of .R.

JPGS

Displays pages stolen from address space.

Type:

Minor of address space majors

To display this value as a rate, use an argument of .R.

JPUI

Displays address space page-ins.

Type:

Minor of address space majors

To display this value as a rate, use an argument of .R.

JPUO

Displays address space page-outs.

Type:

Minor of address space majors

To display this value as a rate, use an argument of .R.

JRCM

Displays composite job recommendation value.

Type:

Minor of address space majors

This value is calculated as:

$$\text{JRCM} = \text{JRCP} + \text{JRIO} + \text{JRWL} + \text{JRST}$$

At certain times, the numbers may not add up to JRCM due to SRM algorithm adjustments.

SUAL

Displays total service units since the last swap-in.

Type:

Minor of address space majors

This value is calculated as:

$$\text{SUCP} + \text{SUIO} + \text{SUMS (All service units)}$$

The service units for nonswappable address spaces are calculated about every 20 seconds. Keep this in mind especially when you use the .R rate argument.

SUCP

Displays CPU service units since the last swap-in.

Type:

Minor of address space majors

The service units for non-swappable address spaces are calculated about every 20 seconds. Keep this in mind especially when you use the .R rate argument.

SUIO

Displays I/O service units since the last swap-in.

Type:

Minor of address space majors

The service units for non-swappable address spaces are calculated about every 20 seconds. Keep this in mind especially when you use the .R rate argument.

SUMS

Displays Main Storage Occupancy (MSO) service units since the last swap-in.

Type:

Minor of address space majors

The service units for non-swappable address spaces are calculated about every 20 seconds. Keep this in mind especially when you use the .R rate argument.

SUPR

Displays service units at beginning of period.

Type:

Minor of address space majors

The service units for non-swappable address spaces are calculated about every 20 seconds. Keep this in mind especially when you use the .R rate argument.

SWPR

Displays swap reason.

Type:

Minor of address space majors

Swap reasons can be any of the following:

Exchange

Exchange swaps (control domain MPL)

Unilatrl

Unilateral swaps (control system MPL)

Long-Wait

Long wait (address space asked to wait more than 0.5 second)

Det-Wait

Detected wait (address space was found waiting)

Term-In

Terminal input wait (TSO waiting on user input)

Term-Out

Terminal output wait (TSO waiting for TCAM/VTAM output buffers)

Aux-Strg

Swapped to correct a shortage of auxiliary storage

Real-Str

Swapped to correct a shortage of real storage

Request

Swapped by SRM for system activity (for example, reconfigure storage to accommodate real storage address spaces)

Enq-Xchg

Swapped to expedite the processing of an enqueue by its owner

Transwap

Swapped while in transition from swappable to nonswappable status

Cent-Str

Swapped out to improve central storage availability

Sys-Page

Swapped out to reduce the system page fault rate

Too-Long

Swapped out to enable the swap-in of an address space that had been swapped out too long

APPC Wt

Voluntary swap-out of an idle APPC address space

If the last letter of the wait reason is an asterisk (*), the address space is ready to do work, but the SRM did not yet swap it back in. For example, if the system is busy and users press Enter, they are

ready to do work. But since the system is so busy, the SRM did not have time to swap them back in. Their swap reason code is Term-In*.

Address space time information

ELAP

Displays elapsed time for address space.

Type:

Minor of address space majors

The display uses one of the following formats:

sss

SEC (seconds)

mm:ss

MN (minutes)

hh:mm

HR (hours)

dd:hh

DY (days)

For example:

```
BATJ  USER21A  DUMPJOB
ELAP  25 SEC   5:39 MN
```

THNK

Displays TSO think time.

Type:

Minor of address space majors

TLTW

Displays time since last terminal wait.

Type:

Minor of address space majors

TMCP

Displays time since last CPU usage evaluation.

Type:

Minor of address space majors

TMIO

Displays time since last I/O usage evaluation.

Type:

Minor of address space majors

TMLA

Displays time since last analysis by SRM.

Type:

Minor of address space majors

TMLR

Displays length of last transaction.

Type:

Minor of address space majors

TMPD

Displays time in current period.

Type:

Minor of address space majors

TMSW

Displays time since last swap.

Type:

Minor of address space majors

TMTR

Displays time in transaction.

Type:

Minor of address space majors

This field displays the time since the last transaction began. Normally, TMTR measures the total job time. Under certain circumstances, however, TMTR might measure only the time since a job step change.

TMWL

Displays time since last evaluation by Workload Manager.

Type:

Minor of address space majors

WAIT

Displays time address space has been waiting.

Type:

Minor of address space majors

WATL

Displays step wait time limit.

Type:

Minor of address space majors

Miscellaneous

ASTG

Displays address space threshold group.

Type:

Minor of address space majors

For different jobs, the appropriate threshold for each exception may be extremely different. To give you the necessary flexibility, OMEGAMON lets you define a set of address space threshold groups using the ASG immediate command.

Each address space threshold group contains a set of jobnames.

Sometimes OMEGAMON cannot discover why a job is waiting. It may be that it is *supposed* to wait until some other event occurs. The WAIT exception produces a warning message for these jobs, unless they are assigned to an address space threshold group with SKIPUW set equal to Yes.

If there is no address space threshold group defined for this job, None appears.

To display a description of the first group that appears (reading from left to right), place a D in the label field. The description shows:

- the group ID (User ID)
- whether SKIPUW is in effect or not (If SKIPUW is on, disabled waits *are* to be reported for the WAIT exception)
- a list of all address space exceptions in effect for this group
- each exception's threshold, color intensity, and bell status (ON or OFF)

For example:

```

JOBN   PCAUTH
DASTG
+      Address Space Threshold Group 2
+      Name = ('GRS', 'PCAUTH', 'TRACE', 'CONSOLE', 'ALLOCAS')
+      SKIPUW = No
+      ELAP = 2592000 ---- ----- WAIT = 2592000 ---- -----
+      NVSC =      500 ---- ----- VISC =      500 LVL2 BELLON-
+      WSLO =      4 ---- ----- WSHI =    1024 ---- -----
+      FXFR =      50 LVL1 BELLON- SWPC =      2 ---- -----
+      PSCU =      90 ---- -----
+      -----

```

IODP

Displays I/O dispatching priority.

Type:

Minor of address space majors

IOJ

Displays I/O count.

Type:

Minor of address space majors

You usually use this command with .R to produce the EXCP rate.

To display the value as a difference between two OMEGAMON cycles, use an argument of .D.

JSTA

Displays job status.

Type:

Minor of address space majors

The output for this command is *aaabccc*, where

aaa

dispatchability:

CPU

Address space is CPU dispatchable

WAT

Address space is waiting

I/O

Address space has performed I/Os but is currently not CPU-dispatchable

b

transaction flag:

*

Address space is in a transaction

b

Address space is not in a transaction

ccc

location:

RES

Address space is resident

NSW

Address space is resident and non-swappable

LSW

Address space is logically swapped

SWP

Address space is swapped out

DLY

Address space is delayed

LGID

Displays TSO logon ID that submitted a job.

Type:

Minor of address space majors

LGID is not valid for started tasks.

QCB

Displays enqueues owned by address space.

Type:

Minor of address space majors

Format:

QCBnnn

The variable *nnn* suppresses the first *nnn* lines.

If you enter an X in the argument field (QCB X), it suppresses any enqueue whose major name is in the Enqueue Exclusion Table that the XQN immediate command defines. The default setting suppresses all data set enqueues (SYSDSN).

A + after the minor name indicates truncation. An S in the label field limits the display of minor names to 43 bytes.

To see all enqueues and reserves, use the XQCB immediate command.

SWCT

Displays short wait count.

Type:

Minor of address space majors

Often short waits result when an address space performs an I/O and is required to wait on it (nonbuffered files, for example).

You usually use SWCT with .R to produce the rate of short waits per second.

SWPC

Displays transaction swap count.

Type:

Minor of address space majors

This is the swap count for a batch job, STC, or TSO transaction.

The count contains both physical and logical swaps.

TRCT

Displays transaction count for swapped in address spaces.

Type:

Minor of address space majors

TRXC

Displays transaction class name.

Type:

Minor of address space majors

TRXN

Displays transaction name.

Type:

Minor of address space majors

Chapter 11. Authorized commands

Authorized commands are sensitive in nature because they can alter data in storage and can display data from restricted storage areas. Misuse of authorized commands can jeopardize your system and the integrity of your data. Use them with caution.

The OMEGAMON internal security facility provides security for all OMEGAMON commands. Each OMEGAMON 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 lower degrees of protection. A setting of 0 means that any user can access the command.

OMEGAMON allows you to secure OMEGAMON commands either internally or by setting up an interface with your installation's external security system (for example, RACF or ACF2). For more information about the security facility, see [Authorize users to issue Take Action commands](#).

The term *authorized commands* refers to certain sensitive or powerful commands that IBM ships with a security level of 3. These commands can be accessed only by entering a password. We ship all other commands with a security level of 0.

This section gives introductory information about using authorized commands, and then describes the commands and their minor commands (see [Table 11 on page 225](#) for a complete list of the authorized commands). The minor commands of the authorized commands discussed here are not authorized.

Using authorized commands

This section explains basic information about using the authorized command facility, such as system access considerations, supplying the action character, accessing other address spaces, and specifying addresses.

Supplying the password

If you are using external authorization, to access authorized commands you need to be assigned to Level 3 authority by your external security system.

If you are using internal security, you need to enter your password via the /PWD INFO-line command. For information on how to use /PWD, see [Chapter 3, "Operational commands," on page 35](#).

How to use the action character

You must enter most authorized OMEGAMON commands with an action character in column 1 before OMEGAMON can execute the command. The action character is a hyphen (-). Some commands require this character whenever you invoke them; others require the action character only on certain occasions.

If you omit the action character from a command that requires it, OMEGAMON 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 then executes the command.

[Table 11 on page 225](#) shows the authorized commands that require the action character, and shows when they require it.

Command	Action character required
.DSA	No
APFU	Yes (when using an option that alters the APF list)
ALI	No

Table 11. Commands that require the action character (continued)

Command	Action character required
ALIB	No
CONS	No
CONU	No
CSAF	No
FNDU	Yes
KILL	Yes
LPAM	No
MCHN	No
MCTL	Yes
MDEF	No
MLST	No
MSCN	No
MZAP	Yes
OCMD	Yes
OSPC	No
PEEK	Yes (when collecting new data)
RCMD	Yes
SCHN	Yes
SLST	Yes
SSCN	Yes
SZAP	Yes
XMCH	Yes
XMLS	Yes
XMSC	Yes
XMZP	Yes

OMEGAMON SRB/XMS facility

OMEGAMON has several authorized commands that must access other address spaces to perform their function (for example, FNDU, PEEK, XMLS, XMSC, and XMZP). Generally, OMEGAMON uses cross memory services (XMS) to access nonswappable address spaces and SRB routines to access all other address spaces.

Note: OMEGAMON completely recovers from all abnormal conditions; it should never cause a system dump or affect the target user in any way (other than to swap it into memory).

SRB time-outs

After OMEGAMON schedules an SRB, it waits for the SRB to signal completion via a cross-memory POST. Until the SRB routine dispatches and executes, the OMEGAMON-authorized command waits to POST.

Under normal circumstances, this is no problem, because SRBs run at extremely high priority within z/OS and complete very quickly.

Before dispatching an SRB into an address space, however, OMEGAMON must first swap the user into memory. The act of scheduling an SRB marks the address space as having ready work. While this means that the SRM eventually brings the user in, it is no guarantee that the swap-in will take place promptly. If the user is in a high-contention domain or in a domain with a maximum MPL of 0, the swap-in may not occur within a reasonable period of time.

To protect against this situation, OMEGAMON schedules a time-out after 10 seconds. If the SRB does not dispatch in this period, OMEGAMON purges it automatically and continues. If this occurs, OMEGAMON displays the following message.

```
WARNING      SRB TIMED OUT AFTER 10.00 SECONDS (RC=8)
```

This message indicates that z/OS could not service OMEGAMON SRB within the 10 second interval, so the SRB aborts rather than continuing to wait. Because the problem may only be temporary, we suggest you retry whatever command causes the condition.

You can also use the SWPI command to try to force the user into memory. See [“Address space minor commands”](#) on page 261 for cautions about SWPI.

SRB/XMS routine program checks

If the target address space is damaged, some of the target user’s control blocks may be overlaid, or may contain invalid pointers or addresses. OMEGAMON monitors this condition by checking the ID field of each control block before use. If it finds a suspect control block, it aborts the analysis, and displays an error message.

In some cases, this situation can cause the SRB/XMS routine to program check when it runs in the target address space. OMEGAMON SRB/XMS routines recover via a functional recovery routine (FRR) when a program check occurs, and the issuing command displays a warning message such as the following one.

```
WARNING      SRB ROUTINE TERMINATED ABNORMALLY - RC=S0C4
PSW AT TIME OF ERROR = xxxxxxxxxxx xxxxxxxxxxx
+ 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
```

This hex display shows the PSW and general registers at the time of the program check.

ATTN or PA1 to abort SRBs

It is possible in TSO mode to press ATTN or PA1 to interrupt OMEGAMON while it waits on an SRB completion. Under normal circumstances, the SRB commands execute so quickly that pressing ATTN has no effect. But in the event OMEGAMON hangs on an SRB and the 10 second timeout does not occur, you can purge the SRB prematurely.

If you purge the SRB prematurely, the following message may appear:

```
WARNING      SRB ABORTED BY ATTENTION (RC=12)
```

While you can interrupt, it is much better to let OMEGAMON timeout normally after 10 seconds.

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. An address consists of an anchor, optional modifiers, and an optional predefined name.

Anchors

An anchor is the base address of an address specification. It can be:

absolute

The hexadecimal address.

symbolic

Up to eight alphanumeric characters, including @, #, and \$. See [“Predefinitions” on page 228](#).

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 addressing. Use the percent symbol (%) for 24-bit (370) addressing. Use an exclamation point (!) as the symbol for 64-bit addressing.

This modifier indicates that the location pointed to is itself an address.

You can use these modifiers to create very powerful and versatile address expressions. For example, the following address expression locates the TIOT of the currently executing z/OS task.

```
10%%+4%+C%
```

10%

Treats the data at location X'10' as a 24-bit address. This is the address of the z/OS CVT.

%

This 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 double word (8 bytes). The double word contains:

- the address of the next TCB to dispatch (bytes 0 to 3)
- the address of the currently dispatched TCB (bytes 4 to 7)

+4%

Adds X'04' to the address of the double word, and then treats the data at the displacement as a 24-bit address. This 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 is the address of the TIOT.

Predefinitions

You can specify or reference an address by a name consisting of up to eight alphanumeric characters, including @, #, and \$. The MDEF command specifies address names.

MDEF

Defines names for addresses.

Type:

Immediate

Format:

```
MDEFcc 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 you want to name.

name
The address name. Up to eight alphanumeric characters, including @, #, and \$.

Consider the following points:

- OMEGAMON saves the names 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 provides a table of predefined names, but if you define an address with one of the predefined names, OMEGAMON uses the address you define during the session.
- OMEGAMON places a comment character (>) in front of MDEF after it executes.

In the next screen display, MDEF gives the name XYZ to the address at offset X'4A' in the CSA.

```
MDEF CSA+4A,XYZ
```

Displaying and modifying storage

This section describes the authorized commands that display, scan, or modify address names for storage locations.

Storage display commands

The following commands display storage locations.

MLST

Displays bytes of memory from the common area or the OMEGAMON private storage area.

Type:

Immediate

Format:

aMLSTc addr,dlen

a

K in the label position requests a display showing the fetch protection key for each virtual block in the range specified and whether fetch protection is ON or OFF.

c

Specifies the format of the output.

B or b

hex and character (default)

C

character only

X

hex only

addr

The first address of storage that OMEGAMON displays.

dlen

The number (up to eight hex digits) of bytes that OMEGAMON displays. The default is 16 (X'10') bytes.

In the following screen display, MLST lists 32 (X'20') bytes starting at address 1EB0 in character format.

```
MLSTC 1EB0,20                               Addr= 00001EB0
+0000 *.....h.... .....167.....*
```

In the next screen, 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... ..*
```

The next example shows MLST with K in the label field, which displays fetch protection information.

```
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
```

XMLS

Displays storage from z/OS address spaces.

Type:

Immediate

Format:

aXMLSc targ,addr,dlen

a

An action character in column 1.

-

Changes to a comment character (>) after command executes.

<

Does not change to a comment character after command executes. Use this action character to repeat the command.

c

The format of the output.

B or b

hex and character (default)

C

character only

X

hex only

targ

The target z/OS address space. It can be:

nnnn

decimal ASID number

ccccccc

jobname

addr

The first address of storage that OMEGAMON displays.

dlen

The number (up to eight hex digits) of bytes that OMEGAMON displays. The default is 16 (X'10') bytes. The maximum is 4096 (X'1000') bytes.

The following figures show uses of the XMLS command.

In the following screen, XMLS displays 32 (X'20') bytes from address space 21 (starting at address 1EB0) in character format.

```
>XMLSC 21,1EB0,20
>storage at 00001EB0 ASID=21:
> 0000 *ABCDEFGH IJKLMNOP QRSTUVWX Z0123456 *
```

In the next screen, XMLS displays 16 (X'10') bytes from the target z/OS address space 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 *ABCDEFGH IJKLMNOP*
```

In the next screen, XMLS displays 16 (X'10') bytes from address space 21, starting at FF32D6 in hex and character format. The less-than symbol (<) prevents OMEGAMON from commenting out the command.

```
<XMLSB 21,FF32C1+15,10
```

Typical output for the XMLS command with the less-than symbol (<) is shown here.

```
<XMLSB 21,FF32C1+15,10
+storage at 00FF32D6 in MVS ASID=21:
+ 0000 4AA800F7 D3700000 00000000 000000F8 *.y..7L.....8*
```

Storage scan commands

The following commands scan storage locations for a specified string of values and display the string if it is found.

MCHN

Scans tables in the common area or in the OMEGAMON 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 displays the table element that contains the string.

Use MCHN to examine:

- Common Storage Area (CSA)
- System Queue Area (SQA)
- nucleus

If you want to search private storage areas other than OMEGAMON storage areas, use the XMCH command.

Format:

MCHNc addr,string,olen,chain,dlen

c

The format of the output.

B or b

hex and character (default)

C

character only

X

hex only

addr

The address of the first table element that OMEGAMON scans.

string

The hex string OMEGAMON uses for the scan. If you enclose this argument in single quotes, OMEGAMON assumes it is a character string.

Note: OMEGAMON interprets two single quotes (") within a character string as a single quote (').

olen

The offset (in hex bytes) to the string in the table element; the comparison starts at this point. You may 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 may precede chain with a plus sign (+) or minus sign (-).

dlen

The number of bytes (up to eight hex digits) that OMEGAMON displays if the scan is successful. The display starts at the beginning of the string. The default is 16 (X'10') bytes.

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:

- 0
- -1
- addr (the table is a ring)

The next screen display 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 appears if the scan is successful.

```

                                     Addr=007DA000
MCHN AAB6C8,D6C30199,8,4
+ 0000 E2E2C3E3 00000000 D6C30199 00000000 *SSCT  0C  r  *
```

MSCN

Scans storage for a string of data and displays the location.

MSCN scans the address space in which OMEGAMON resides for a string of hex or character values. If the scan is successful, OMEGAMON displays the string.

Type:

Immediate

Format:

MSCNc addr,string,slen,dlen

c

The format of the output.

B or b

hex and character (default)

C

character only

X

hex only

addr

The first address of storage that OMEGAMON scans.

string

The hex string OMEGAMON uses for the scan. If you enclose it in single quotes, OMEGAMON assumes it is a character string.

Note: OMEGAMON interprets two single quotes (") within a character string as a single quote (').**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 screen display, MSCN scans the first 1000 bytes of the TIOT entry for the character string OMHELP (see [“Address specification for storage commands” on page 228](#) 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%, 'OMHELP', 1000, 14
```

Typical output of the MSCN command is shown here.

```
MSCN 10%%+4%+C%, 'OMHELP', 1000, 14                               Addr=0061701C
+ 0000 D6C3C8C5 D3D74040 60BCA000 80001B00 *OMHELP -.....*
+ 0010 14010100 *.....*
```

XMCH

Scans tables in the z/OS address space.

XMCH scans the elements of a table for a string of hex or character values. If the scan is successful, OMEGAMON displays the table element that contains the string.

Use XMCH to search z/OS address spaces. Use MCHN to search the address space in which OMEGAMON resides.

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 executes.
- < Does not change to a comment character after command executes. Use this action character to repeat the command.

c

The format of the output.

B or b

hex and character (default)

C

character only

X

hex only

targ

The target z/OS address space. It can be:

nnnn

decimal ASID number

ccccccc

jobname

addr

The address of the first table element that OMEGAMON scans.

string

The hex string OMEGAMON uses for the scan. If you enclose it in single quotes, OMEGAMON assumes it is a character string.

Note: OMEGAMON interprets two single quotes (") within a character string as a single quote (').

olen

The offset (in hex bytes) to the string in the table element; the comparison starts at this point. You may 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 may precede chain with a plus sign (+) or minus sign (-).

dlen

The number of bytes (up to eight hex digits) that OMEGAMON displays if the scan is successful. The display starts at the beginning of the table element. The default is 16 (X'10') bytes.

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:

- 0
- -1
- addr (the table is a ring)

The next screen display shows a typical XMCH command.

```
-XMCH USER14,AAB6C8,D6C30199,8,4
```

In this example, XMCH scans a table in the target z/OS address space 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 appears if the scan is successful.

```
>XMCH USER14,AAB6C8,D6C30199,8,4
>Storage at 007DA000 in USER14 ASID=21:
> 0000 E2E2C3E3 00000000 D6C30199 00000000 *SSCT 0C r *
```

XMSC

Scans storage in the z/OS address space for a string of data and displays the location.

XMSC scans a z/OS address space for a string of hex or character values. If the scan is successful, OMEGAMON displays the string.

Type:

Immediate

Format:

aXMSCc targ,addr,string,slen,dlen

a

An action character in column 1.

-

Changes to a comment character (>) after the command executes.

<

Does not change to a comment character after the command executes. Use this action character to repeat the command.

c

The format of the output.

B or b

hex and character (default)

C

character only

X

hex only

targ

The target z/OS address space. It can be:

nnnn

decimal ASID number

ccccccc

jobname

addr

The start address of the scan.

string

The hex string OMEGAMON uses for the scan. If you enclose it in single quotes, OMEGAMON assumes it is a character string.

Note: OMEGAMON interprets two single quotes (") within a character string as a single quote (').

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 string. The default is 16 (X'10') bytes.

In the next screen display, XMSC scans 1000 bytes in the target z/OS address space specified by the jobname USER14, starting at location 515988 for the character string WORKAREA. If the scan is successful, OMEGAMON displays 14 hex bytes in hex and character format starting at WORKAREA.

```
-XMSC USER14,515988,'WORKAREA',1000,14
```

The following output appears if the scan is successful.

```
>XMSC USER14,515988,'WORKAREA',1000,14
>storage at 00515988 in USER14 ASID=21:
> 1B8 E6D6D9D2 C1D9C5C1 00000000 000C0000 *WORKAREA.....*
> 1C8 000C002C *.....*
```

Storage modification commands

These commands modify storage locations. Use them with caution.

MZAP

Modifies the contents of the common area or of the OMEGAMON private storage area.

Type:

Immediate

MZAP modifies the contents of the common area:

- Common Storage Area (CSA)
- System Queue Area (SQA)
- Nucleus

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 address space for debugging purposes.
- 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 this is necessary, OMEGAMON 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 executes.

addr

The address of the string that OMEGAMON may modify.

ver

The verify string. OMEGAMON modifies storage only if OMEGAMON finds this string at *addr*. If OMEGAMON does not find string, it displays what is actually at *addr*.

rep

The replacement string. If OMEGAMON finds *ver* at *addr*, *rep* replaces *ver*.

Note: The verify and replacement strings must be the same length.

In the next screen display, MZAP changes a fullword at location 6764 from X'A' to X'64'.

```
MZAP 6744+20,0000000A,00000064
```

The next screen shows how MZAP changes an 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 target address space.

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 executes.

targ

The target z/OS address space. It can be:

nnnn

decimal ASID number

ccccccc

jobname

addr

The address of the string that OMEGAMON may modify.

ver

The verify string; OMEGAMON modifies storage only if OMEGAMON finds this string at *addr*. If OMEGAMON does not find the string, it displays what is actually at *addr*.

rep

The replacement string. If OMEGAMON finds *ver* at *addr*, *rep* replaces *ver*.

Note: The verify and replacement strings must be the same length.

In the next screen display, XMZP changes a byte at location A0160 in the master scheduler address space.

```
-XMZP *MASTER*,A0160,0A,64
```

In the next screen display, XMZP changes a halfword at C4834 in the PRODJOB address space from X'1854' to X'0700'.

```
-XMZP PRODJOB,C4834,1854,0700
```

Displaying and modifying data space and hiperspace storage

OMEGAMON provides commands to display or modify z/OS data space and hiperspace storage. The commands in this section parallel the functions of the MLST, MCHN, MSCN, and MZAP storage commands.

Because of the potential security risk associated with using these commands, the .DSA command exists to provide an extra level of protection.

.DSA

Sets and displays authorization to list or zap nonshareable 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 have been defined by the owner as able to be shared by other address spaces).

Command operands:**ON**

Turns on data-only space authorization, (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 .DSA with no operand displays the current status of data-only space authorization.

OSPC

Lists the attributes of the owner of a data-only space.

Type:

Immediate

Format:

OSPC *spacename*

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 will display any space names beginning 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.

```
>OSPC MYSPACE
+ TYPE          ASID      JOBNAME      TCB address  SPACE NAME
+ Data space    12        MYJOB        007FFA10     MYSPACE
+ Hiperspace    22        HISJOB        007B7CB0     MYSPACE
```

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 executes.

<

Does not change to a comment character after command executes. Use this action character to repeat the command.

c

Specifies the format of the output:

B or b

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.

addrThe starting address of the data. See [“Address specification for storage commands”](#) on page 228 for additional options on specifying addr.**plen**

The number (1 to 8 hex digits) of bytes to print. The default is 16 (X'10') bytes or 1 line.

SLST will list memory from data-only spaces. When necessary, an SRB will be 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*
```

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 executes.

<

Does not change to a comment character after command executes. Use this action character to repeat the command.

c

Specifies the format of the output:

B or b

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 228](#) for additional options on specifying addr.

string

The comparison string for the scan. Either a hexadecimal string or a character string. Use single quotes around a character string. Do not use quotes 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 will scan data-only space storage until a match to the string is found or the length of storage specified is exhausted. When necessary, an SRB will be scheduled into the address space of the TCB owning the data-only space to be scanned.

Following 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*
```

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 located 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 executes.

<

Does not change to a comment character after command executes. Use this action character to repeat the command.

c

Specifies the format of the output:

B or b

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 228](#) for additional options on specifying addr.

string

The comparison string for the scan. Either a hexadecimal string or a character string surrounded by single quotes.

off1

The offset from the beginning of the control block to the location of the comparison string. This value may 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 may 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 will be scheduled into the address space of the TCB owning the data-only space to be scanned.

This example shows the SCHN display for a data space.

```
>SCHN MYJOB,MYSPLACE,1000,'TEST',0,30,20
+Storage at 00001100 in dataspace MYSPLACE, job MYJOB ASID=12
+ 0000 E3C5E2E3 40C4C1E3 C140E2D7 C1C3C540 *TEST DATA SPACE *
+ 0010 F0F1F2F3 F4F5F6F7 F8F9C1C2 C3C4C5C6 *0123456789ABCDEF*
```

SZAP

Modifies the contents of data-only space storage.



CAUTION: There is a potential integrity exposure when using SZAP on hiperspaces. SZAP will use HSPSERV to read in a page of data from the target hiperspace, check the data, alter the data, and finally 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 could be lost. There is no available enqueue mechanism to guard against this exposure.

Type:

Immediate

Format:

-SZAP jobname, spacename, addr, vercode, repcode

-

Action character 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 228](#) for additional 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 will be scheduled into the address space of the TCB owning the data-only space to be zapped. Here is an example of using SZAP.

```
>SZAP MYJOB,MYSPACE,1000,00000005,00000008  
> >> OB7110: Memory Zap Successful <<
```

Console-related commands

The commands discussed in this section relate to console operations.

OCMD

Issues z/OS and JES2 operator commands from an OMEGAMON terminal.

Type:

Immediate

Format:

-OCMD nn [CONS={conid|conname}] ccccc or -OCMD nn [CONX=xconid] ccccc

-

Required action character. This changes to a comment character (>) after the command executes.

nn

For commands that accept a return destination, nn indicates which operator's console will receive the response. If you omit nn , the response goes to the master console.

conid

Specifies which operator's console issued the command. Overrides the nn operand.

conname

Specifies the console name from which the command originated.

xconid

Specifies which operator's console issued the command in 1-8 hexadecimal digit format. Overrides the nn operand.

cccc

Indicates an or JES command.

OMEGAMON issues the command (cccc) that you supply by using SVC 34.

```
-OCMD01 SEND 'PLEASE RELEASE ALL HELD DATA SETS',USER=TS0001  
-OCMD01 $HJ123  
-OCMD01 C JOB123,DUMP
```

OMEGAMON 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.

In TSO mode, you can request OMEGAMON to return the output of a command back to your terminal. To do this, issue OCMD with a console ID of 99. For example:

```
-OCMD99 D T
```

displays the date and time on your TSO terminal.

Some z/OS commands do not send output to TSO users, even though you use OCMD99. This is because OCMD can only request the z/OS command scheduler to send the output back to the OMEGAMON TSO session; some z/OS components do not honor this request. The JES2 \$ commands are an example of this restriction. For example, the output of

```
-OCMD99 $DA
```

appears on the master console, not your TSO session.

Important: OCMD99 does *not* work for any OMEGAMON session type except TSO. In particular, it does not work for dedicated or VTAM sessions.

RCMD

Routes z/OS and JES2 commands to a desired system in a Sysplex.

Type:

Major

Format:

```
-RCMDnn sysname [CONS={conid|conname}] ccccc -RCMDnn sysname  
[CONX=xconid] ccccc
```

OMEGAMON issues the command (cccc) that you supply via SVC 34.

RCMD requires the action character (-) in column 1. The action character changes to a comment character after execution.

nn indicates which operator's console will receive the response. If you omit *nn*, the response goes to the master console.

sysname specifies the system name where the command executes.

cccc is a z/OS or JES2 command.

conid specifies which operator's console issued the command. This operand overrides the *nn* operand.

xconid specifies which operator's console issued the command in 1-8 hexadecimal digit format. 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}] CONSnn [CONX=xconid]
```

nn is a z/OS console number. This number ranges from 01 to the maximum number of consoles generated for your installation. If *nn* is omitted, OMEGAMON 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.

xconid selects the z/OS operator's console by hexadecimal number. ID may be 1 through 8 hexadecimal digits. This operand overrides the *nn* operand.

CONS displays three types of information on 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.

The CONS minor commands enable you 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 61 on page 244](#) for an example.)

Note that OMEGAMON accesses the CONSOLE address space 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 61 on page 244](#) is an example of an OMEGAMON 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*  IP01
+       - 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 TS0003G ENDED
+       -           $HASP309  INIT 3 INACTIVE ***** C=I
+       - JOB 1111 $HASP250 TS0003G IS PURGED
+       - STC  721 DFS2500I *MDA00  IP01
+       - DATABASE BE3PARTS SUCCESSFULLY ALLOCATED
+       - STC  721 DFS2500I *MDA00  IP01
+       - DATABASE BE3PSID1 SUCCESSFULLY ALLOCATED
+       | JOB 1114 @83 DFS3125A PRIMER SAMPLE TEST, REPLY CONT, LOOP, ABEND, 0
+       | CANCEL JOB
+       - TSU 1115 $HASP100 TS0045  ON TSOMEGAMON/IMSNRDR
+       - TSU 1115 $HASP373 TS0045  STARTED
+       00      IEE600I REPLY TO 83 IS;ABEND
=====

```

Figure 61. OCMD and CONS Output

CONU

Locates the output buffer for a z/OS operator console by device address.

Type:

Major

Format:

CONU xxx

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 automatically locates the output buffer for the master console. CONU displays the address and the console ID:

```
CONU 6A0 Master Console 6A0 ( 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, to display commands that require operator action for console 3, enter the following command:

```
CONS03 Console 6A0 (ID=3)
ACTN
```

LINE

Displays the last *nn* lines from the console you select.

Type:

Minor of CONS and CONU

Format:

LINEnn

To display the last 12 lines of the screen of console 11, enter the following command:

```
CONS11  Master Console (ID=11)
line12
```

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
```

For JES2, OMEGAMON requires the \$TM command to change the status information routing.

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)
mmt
+ 2000 09.07.47 JOB 2225 *13 IEC701D M 370,VOLUME TO BE LABELED CA1759
```

Collecting data about address spaces

The PEEK major command collects information about a single address space. Its associated minor commands TBD.

PEEK

Collects information about a single address space.

Type:

Major

After you issue the PEEK command to collect information from the target address space, you can format and display this information with various PEEK minor commands.

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 be re-executed on succeeding cycles.

┌

Enables minor commands to execute with previously collected data.

targ

The target address space. It can be:

ccccccc
jobname
nnnn
decimal ASID number

OMEGAMON address space

For example, to gather data from a job named PAYROLL (with an ASID of decimal 25), enter:

```
-PEEK PAYROLL
```

or

```
-PEEK 25
```

PEEK accesses the PAYROLL address space, removes the action character from column 1, and displays the following:

```
PEEK PAYROLL ASID=25 >> 0B8112: Data Collection Initiated <<
```

When PEEK 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 you issue PEEK with the action character in the label field, PEEK collects current information. If you issue PEEK without an action character, it uses the data in the work area from the previous update.

Note that PEEK 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. Since SRBs run at the highest priority and increase the swapping load, overuse can degrade performance. The .SET command contains two keywords to set OMEGAMON 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 to 1000000.

The PEEK command traces control block chains. If OMEGAMON encounters a damaged target address space, some of the control blocks examined may have chained into a loop, and OMEGAMON issues a warning message. The warning may also appear when you chain through an address space that has a complex TCB structure. In this case, the cause may 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 33553408.

The first time you use PEEK in an OMEGAMON session, it obtains a work area (32K by default) from the private area to hold the collected data. OMEGAMON gives you a warning message if the data does not fit within the work area.

Issue .SET and increase the value of PEEKSIZE= to increase the work area size. Then reissue the PEEK command and it will collect the data. You can save the new PEEKSIZE definition in a user profile.

The following minor commands can be used with PEEK:

AMAP

Displays a map of virtual storage utilization within the private area.

Type:

Minor of PEEK

The AMAP minor command of PEEK 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 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>
amapB <map virtual storage below the 16M line>
```

Figure 62 on page 247 shows the AMAP display format.

```
PEEK K3RCOL22 ASID=323, collected at 15:32:26
amap
+ ===== 2 Gig Line ===== <== 7FFFFFFF Highest 31-bit address
+ |-----| <== 7FFFFFFF Top of Extended Private
+ |//////| >----- 8K ELSQA/SWA unallocated
+ |//// System Area ////| >----- 134K Fragmented free space
+ |//////| <== 7F654000 Current bottom of ELSQA/SWA
+ |-----|
+ | Available | >----- 0K Avail. for ELSQA/SWA only
+ |-----| <== 7F653FFF Extended User Area Limit
+ | Available | >----- 1471M Avail. for ELSQA/SWA/USER
+ |-----| <== 236F8FFF Current top of Ext. User Area
+ |//////| >----- 120K Largest free block
+ |//// User Area ////| >----- 120K Extended User unallocated
+ |//////| >----- 144K Fragmented free space
+ |-----| <== 23400000 Bottom of Extended Private
+ ===== 16 Meg Line ===== <== 00FFFFFF Highest 24-bit address
+ |-----| <== 007FFFFFF Top of Private
+ |//////| >----- 0K LSQA/SWA unallocated
+ |//// System Area ////| >----- 100K Fragmented free space
+ |//////| <== 007C2000 Current bottom of LSQA/SWA
+ | Available | >----- (Below User Area Limit)
+ | Available | >----- 0K Avail. for LSQA/SWA only
+ |-----| <== 007C1FFF User Area Limit(Was 007FFFFFF)
+ | Available | >----- 6956K Avail. for LSQA/SWA/USER
+ |-----| <== 000F6FFF Current top of User Area
+ |//////| >----- 12K Largest free block
+ |//// User Area ////| >----- 12K User unallocated
+ |//////| >----- 67K Fragmented free space
+ |-----| <== 00006000 Bottom of Private
+ ===== Absolute Bottom ===== <== 00000000 Prefixed Storage Area
```

Figure 62. AMAP display format

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 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 allocated within the extended private area for the extended system area.

Avail. for ELSQA/SWA only

The amount of unallocated 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. Note that the extended system area can allocate storage within this area.

Current Top of Ext. User Area

The highest address currently allocated within the extended private area for the extended user area.

Largest free block

The largest contiguous piece of unallocated 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 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 which are each less than the 4K available for allocation as defined by FQEs.

Current bottom of LSQA/SWA

Lowest address allocated to LSQA/SWA subpools.

Avail. for LSQA/SWA only

Total space available for LSQA/SWA allocation. This includes the LSQA/SWA unallocated 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 unallocated areas within the region used.

Current top of User Area

Highest address 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 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 starting with absolute zero.

DATA

Displays data space and Hiperspace utilization for a given address space.

Type:

Minor of PEEK

The DATA minor of PEEK requires an APF-authorized environment.

Here is an example of the DATA minor display.

```
-PEEK TS0X07 ASID=48 >> OB8112: Data Collection Initiated <<
data Name Type Owing 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
```

If you put an X in the label field, DATA displays extended information.

```
-PEEK TS0X07 ASID=48 >> OB8112: Data Collection Initiated <<
Xdata Name Type Owing Task Current Size Maximum Size
+ SDUMPSWA Basic CR8SPACE (007FE380) 64K 256K
+ Key= 8 Fprot=NO Dref=NO Scope=SINGLE
+ Used= 2660K PageableFrm 2660K FixedFrm= 1648K
+ 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
```

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 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 being given special consideration when the system searches for pages to remove when a shortage arises.

Used

The storage used, in bytes, that is made up of real storage and auxiliary storage backing the data only space.

PageableFrm

The Pageable Real Storage frames, in bytes, that are currently backing the data only space.

FixedFrm

The Fixed Real Storage frames, in bytes, that are currently backing the data only space.

DDNS

Displays information about allocated ddnames.

Type:

Minor of PEEK

Format:

[_]DDNS[nn] [X]DDNS[nn]

—

Displays all ddnames 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.

VOLS

Count of volumes being used by the data set.

PWD

Password protection. A blank indicates that no data set password is in effect.

EXCP

Execute 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

```

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.
- 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  X0IDI   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

```

When the `Programmer Name` field is blank in the job statement, the JOBS minor command does not display that line. JOBCLASS information only appears for batch jobs. The job class is that class originally associated with the job. If a JES command was requeued prior to execution, the original job class from the CLASS= parameter in the JCL JOB statement will appear.

```

PEEK  CONSOLE ASID=5 Job ID: MSTR      Subsystem name: MSTR
JOBS  no job information available

```

LPAG

Displays large page usage information.

Type:

Minor of PEEK

This command displays the following information:

- Number of large memory objects allocated by this address space
- Number of large (1 Mb) pages backed in real storage owned by this address space

LSQA

Displays LSQA (local system queue area) and ELSQA (extended local system queue area) statics.

Type:

Minor of PEEK

The following screen shows a typical LSQA display:

```

----- ZMENU   VTM      OM/DEX   V420./C SP13 07/07/09 18:53:56
PEEK  XCFAS   ASID=6   >> OB8112: Data Collection Initiated <<
lsqa
+           Allocated   Percent   Available   Largest
+           Allocated
+           -----
+ Below 16m line:      207K      2.6%      7920K      7856K
+ Above 16m line:    16486K      .9%      1725M      1725M

```

Figure 63. Typical LSQA results screen

where:

Below 16m line

Indicates LSQA.

Above 16m line

Indicates ELSQA.

Allocated

Is the amount of LSQA or ELSQA allocated.

Percent Allocated

Is the percentage of the total LSQA or ELSQA that is allocated.

Available

Is the amount of LSQA or ELSQA that is available (that is, unallocated). This storage is also available for SWA and USER allocation.

Largest Available

Is the largest area of contiguous LSQA or ELSQA storage that is available (that is, unallocated). This storage is also available for SWA and USER allocation.

MODS

Displays information about the modules currently loaded into the user's jobpack area.

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
```

The name of each module currently in the user's jobpack area appears along with its entry point address, length, use count, and load module attributes. 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 PEEK

The following screen shows an example of the STEP command.

```

PEEK K3RCOL22 ASID=323, collected at 15:32:26
step Job Step Pgm: KOBVTAM; 6 TCBS, 18 Datasets, and 16 Modules
+   Step started at 13:50:20 , now in step # 1 of 1
+
+ For the region below the 16M line:
+
+ Total private region = 8168K   Unused = 6968K
+ Region requested = 0K   Region limit = 8168K
+ Low PVT in use = 952K   Unallocated = 12K   Free = 67K
+ High PVT in use = 248K   Unallocated = 0K   Free = 100K
+ Start of SYSREG: 00002000   End of SYSREG: 00005FFF
+ Start of low PVT: 00006000   End of low PVT: 000F6FFF
+ Current top: 000F6FFF   Limit of region: 007FFFFFFF
+ Start of high PVT: 007C2000   End of high PVT: 007FFFFFFF
+
+ For the extended region above the 16M line:
+
+ Total private region = 1519616K   Unused = 1506796K
+ Region limit = 1519616K
+ Low PVT in use = 2924K   Unallocated = 120K   Free = 144K
+ High PVT in use = 9896K*   Unallocated = 8K   Free = 134K
+ *includes non-freemaintainable system area of 7476K starting at 7F800000
+ Start of low PVT: 23400000   End of low PVT: 236F8FFF
+ Current top: 236F8FFF   Limit of region: 7FFFFFFF
+ Start of high PVT: 7F654000   End of high PVT: 7FFFFFFF

```

ccccccc

The current program name of the topmost JOBSTEP TCB. (This value corresponds to what is in the EXEC statement, unless XCTL was used to transfer control to another load module.) The other counts indicate how many lines of output you may expect from the TCBS, DDNS, and MODS minor commands.

Total private region

The total size of the private area, including areas that can't be allocated.

Region requested

The amount 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. This includes 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 allocated to subpools.

Unallocated

The storage not allocated to subpools.

Free

The storage allocated to subpools but not currently GETMAINED.

SYSREG

An area of storage reserved for use by the region control task.

Current top

The highest allocated address in the low PVT area at the current time.

For the system address spaces (CONSOLE, GRS, ALLOCAS), there is no job information available. In this case OMEGAMON displays:

```

PEEK IEEVWAIT ASID=25, collected at 15:39:39
STEP Region Requested = *N/A*   Region in use =*N/A*   Region Limit =*N/A*
+   Job Step Pgm: IEEVWAIT; 5 TCBS, 6 Datasets, and 2 Modules
+   System Address Space has no steps

```

For the system address spaces, the STEP minor does not display step start time, nor any information available on step number.

SUBP

Displays information about current virtual storage allocations for each storage subpool

Type:

Minor of PEEK

Format:

[_]SUBP[nn] [X]SUBP[nn]

—

Requests information for the subpools allocated to the jobstep TCB and to any TCBs below 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 on 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 or not the X label was entered in front of the SUBP command. Consequently, the totals could be larger than the sums of the private area statistics.

Note that 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:

```

PEEK CPSTEP6A ASID=15, collected at 15:39:39
subp
+ SBP-K   Alloc  Real  #Blks   Addr   Free  #Blks  Mxfree  Program  Own
+ 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
=====

```

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 currently allocated to the subpool (in 4K increments). The storage is not necessarily contiguous.

Real

Amount of real storage backing the virtual allocation in 4K blocks.

#Blks

Number blocks allocated to the subpool.

Addr

Address of the block with the lowest address (hex). In 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 described associated with these subpools.

Own|Shr

Allocations marked SHR are displayed for each TCB that shares the allocation. Allocations marked OWN appear 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 allocated for the address space.

Subtot

Summary of LSQA and user virtual storage allocated.

The final section, which appears 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 allocated to subpool 251, key 8. It is allocated in three blocks and the storage 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 PEEK

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)
=====

```

This is a typical TCB structure for a batch job.

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.

Note that most address spaces on this level point to themselves as the Jobstep TCB.

If you place an A in the label field of TCBS, OMEGAMON displays the actual TCB address under each TCB program name. This information may be useful if several TCBs in the same step invoke the same program.

If you place an X in the label field of TCBS, two extra lines appear 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 appears to indicate which non-dispatchability bit was 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 by far the most common reason for a TCB to be non-dispatchable.

You may optionally specify a 2-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 being PEEKed has many TCBs, you may need to increase the maximum number of control blocks tested with the LOOPCOUNT keyword of the .SET command.

Terminating an address space (KILL)

KILL

Terminates an address space.

Type:

Immediate

KILL terminates an address space, which in turn causes the target address space to terminate with a system A22 abend. Along with the KILL command, you must specify the jobname or the ASID in decimal.

Format:

```
aKILL ccc
```

a

An action character in column 1:

-

Executes the command once and changes to a comment character (>) when the command executes successfully.

ccc

Jobname or ASID

When OMEGAMON responds, you must supply an action character (-) in column 1 before the command executes.

The results of a successful KILL command under OMEGAMON are exactly the same as the results of the z/OS FORCE command; however, there are some differences in how they operate:

- The FORCE command sometimes delays or is unable to execute at all, because it must enqueue on the CSCB chain in order to search it; KILL does not search the CSCB chain, and so avoids this problem.
- FORCE does not operate unless a CANCEL is in effect for the address space; KILL has no such restriction.
- You can only issue FORCE from the z/OS master console; the OMEGAMON security facility controls KILL.

- You cannot use the unit address instead of the job name to cancel a started task.

Important: Sometimes you must remove a job from the system to avoid an IPL. The z/OS FORCE command attempts this, but frequently fails; the KILL command is successful in many cases in which FORCE does not work. However, KILL has no more cleanup than FORCE, so, like FORCE, you should use it with discretion.

Find users of a device (FNDU)

FNDU

Finds all users of a device.

Type:

Major

Format:

aFNDU device

a

An action character in column 1:

-

Specifies that new data be collected from the target address spaces.

b

Enables minor commands to execute with previously collected data.

device

This can be:

xxx

3-digit hexadecimal unit address

cccccc

6-digit volume serial number

If the unit indicated is online, FNDU accesses every active address space, and continues until it finds as many allocations as the UCBUSER field indicates. This can cause OMEGAMON to pause until all address spaces are examined.

For example, if you issue the command

```
-FNDU MVS001
```

or

```
-FNDU 245
```

you might see the following result:

```
FNDU MVS001 245 2 User(s) Holding 3 Allocation(s)
```

This typical FNDU display means that two users are active, and have three data sets allocated on MVS001 between them. To display the specific users, you must use the DSPA minor command of FNDU. If you reissue this command without the action character, it does not display current information. If, for example, you are attempting to determine allocations on paging volumes based on the PAOD exception, be sure to re-enter the action character.

When you try to vary a device offline, the VARY remains pending until all users of the device free their allocations. In the case of a DASD device, it is often difficult to determine which users have the data sets allocated. The z/OS OS command, D U, and the OMEGAMON DALC minor command of DISK, show how many allocations are still held but not who holds them. The z/OS command D U,,ALLOC, displays

the users of a volume but not the ddname or number of current allocations. The authorized command, FNDU, displays more detailed information than DALC or D U,,ALLOC.

FNDU accesses every active address space to look for allocated data sets on the target pack. Because this may involve swapping many users into memory for a short time, FNDU may have significant system overhead. Use it carefully.

Sometimes no allocations of a device are found, but the device use count shows a number greater than zero. This condition occurs in the following cases:

1. At IPL time, the nucleus initialization program (NIP) allocates and opens several system data sets before the standard OPEN and ALLOCATE interface is available. To ensure that the volume is not forced offline, NIP raises the device use count by one for each data set it needs on a volume. Because there are no DD statements for these devices, FNDU does not find them under the master scheduler, and it appears that there is a discrepancy between DALC and FNDU. This affects the SYS1.LOGREC data set, the page and swap data sets, and all of the link list library data sets. Because you don't want to vary these devices offline anyway, this does not present a problem.
2. The other case is more serious. If you use the OS FORCE command or the OMEGAMON KILL command to cancel a user, the Recovery Termination Manager (RTM) does not perform any device allocation cleanup. Devices that the FORCED user allocated do not have their use counts decremented, and so the volumes the job used show allocations that do not exist (and that FNDU cannot find). You cannot vary volumes in this condition offline, unless you use MZAP to zap the UCB USE count (UCBUSER) in memory.

The first time you use FNDU in an OMEGAMON session, it gets a work area to hold the collected data from the target address space. If you use FNDU in dedicated mode, you might have to increase the size of the OMEGAMON V=R region. If the allocation data from all of the active address spaces does not fit within the work area, FNDU issues a warning. Use the WSIZ minor of FNDU to increase the work area size and then reissue the FNDU command.

The following are minor commands of FNDU.

DSPA

Displays details of the data FNDU collected.

Type:

Minor of FNDU

For example:

```
FNDU MVS001 245 2 User(s) Holding 3 Allocation(s)
DSPA DDName Sta,Disp DSName
+ TS0005 SYSUT1 OLD,KEEP SYS3.DATALIB
+ SYS00001 SHR,KEEP SYSCTLG.VMVS001
+ PRODJOB TMPFILE * NEW,PASS TEMPFILE
```

This display indicates that job TSO005 has two data sets allocated on MVS001, and that job PRODJOB has one data set. The asterisk (*) indicates that data set TEMPFILE is currently open.

DSPA accepts a 2-digit argument to suppress the first *nn* lines of output.

DSPO

Displays only those allocations that are currently open.

Type:

Minor of FNDU

DSPC

Displays only those allocations that are currently close

Type:

Minor of FNDU

WSIZ nn

Alters the work area size for FNDU data.

Type:

Minor of FNDU

The variable *nn* is the size of work area in KB.

SEEK contention analysis

SEEK

Displays DASD SEEK contention information for a specified device.

Type:

Major

Format:

```
-SEEK VOLser(vvvvvv) | DEVICE(ccc) INTerval(nn) SAMples(nnn)
```

-

An action character. It is required in column 1 on the first execution of the command, and to force re-collection on any subsequent execution.

VOLser|DEVICE

The volser or device address to be analyzed. It is required whenever the action character is specified in column 1. Volser must be 6 characters long; device can be 3 or 4 characters long.

INTerval

Indicates how frequently SEEK analysis will be performed. Interval is optional with a default of 10 ms. Valid values are 5–500 ms.

SAMples

Indicates the sampling rate as the number of samples taken per interval. Samples is optional, with a default of 70, and a maximum of 999.

An example of the SEEK command follows:

```
-SEEK VOL(OMON28) INTERVAL(100)
```

In the preceding example, SEEK analysis would be performed every 100 milliseconds for 70 samples for volume OMON28.

The following figure displays the output from a SEEK major command:

```
SEEK VOL(OMON28) INT(10)
+ Unit=3DC Volser=OMON28 Samples=70 Interval=010 ms Time=08:43:21
+ Observed Seeks=20 Low Cyl=83 High Cyl=881 Avg Seek=20 cyls
```

Figure 64. SEEK major command output

DATA

Displays detailed data set information for one or all seeks.

Type:

Minor of SEEK

Format:

```
-SEEK VOL(OMON28) INTERVAL(100) XDATA ITEM(nnn | ccccccc | *)
```

nnn

The 1-to 3-digit sample number that detailed information is requested for.

ccccccc

The 1-to 8-character jobname that detailed SEEK information is requested for.

*

Displays detailed data for all observed SEEKS. (default)

X

This is optional, and may be used to display extent information for each extent in a data set associated with a displayed SEEK observation. If not specified, the display includes extent information for only the extent within the data set accessed during the observed SEEK.

The following figure displays the output from a DATA detail display for all observed SEEKS:

```
-SEEK VOL(OMON28) INT(10)
+ Unit=3DC Volser=OMON28 Samples=70 Interval=010 ms SEEKS=20 Time=08:43:21
DATA ITEM(*)
+ # Jobname ASID Dataset name Cyl Ext Low Hi
+ ---
+ 3 TESTJOB1 1C PAYROLL.TEST.FILE 853 1 840 879
+ 6 PRODJOBA 23 PAYROLL.PROD.FILE 83 3 080 099
+ 7 PRODJOBA 23 PAYROLL.PROD.FILE 84 3 080 099
+ 8 XYZJOB 15A XYZJOB.DATA 773 4 773 773
+ 12 TSUSERX 76 TSUSERX.JCL.CNTL 310 2 310 310
+ 15 TESTJOB1 1C PAYROLL.TEST.FILE 854 1 840 879
+ 19 TSUSERY 8D TSUSERY.DUMP.LIST 527 10 520 534
+ 21 TSUSERX 89 TSUSER.COMMON.CLIST 880 1 880 884
+ 28 TESTJOB1 1C PAYROLL.TEST.FILE 855 1 840 879
+ 32 ABCJOB 143 ABCJOB.DATA 395 1 394 396
+ 35 TSUSERX 76 TSUSER.COMMON.CLIST 881 1 880 884
+ 41 PRODJOBA 23 PAYROLL.PROD.FILE 90 3 080 099
+ 42 TSUSERY 8D TSUSERY.DUMP.LIST 528 10 520 534
+ 45 TESTJOB1 1C PAYROLL.TEST.FILE 859 1 840 879
+ 49 TSUSERX 76 TSUSERX.JCL.CNTL 310 2 310 310
+ 53 TSUSERY 8D TSUSER.COMMON.CLIST 881 1 880 884
+ 56 TSUSERY 8D TSUSERY.DUMP.LIST 529 10 520 524
+ 61 TSUSERZ A7 TSUSERZ.MISC.LOG 195 1 195 195
+ 62 XYZJOB 15A XYZJOB.DATA 773 4 773 773
+ 63 PRODJOBA 23 PAYROLL.TEST.FILE 853 1 840 879
```

Figure 65. DATA minor command output for all observed SEEKS

SDAT is a synonym of the DATA minor of SEEK. It can be used when the DATA minor has been secured for the PEEK command, making it unavailable.

PLOT

Graphically displays SEEK contention for a specified device.

Type:

Minor of SEEK

Format:

```
SEEK VOL(OMON28) INTERVAL(100) PLOT
```

The following figure shows the output from a PLOT minor command. The Observed Seeks are plotted by cylinder number. The accessed cylinder is indicated with an asterisk (*) in the graph, along with the direction and distance the device head moved in relation to the previous Seek that was observed during sampling.

```

SEEK VOL(OMON28) INT(10)
+ Unit=3DC Volser=OMON28 Samples=70 Interval=10 ms Time=08:43:21
+ Observed Seeks=20 Low Cyl=83 High Cyl=881 Avg Seek=20 cyls
PLOT
+ # Jobname Cyl 0---100---200---300---400---500---600---700---800--+
+ ---|-----|-----|-----|-----|-----|-----|-----|-----|
+ 3 TESTJOB1 853| | | | | | | | | | | | | | | | | | | | | | | | | | *
+ 6 PRODJOBA 83| *|-----|-----|-----|-----|-----|-----|-----|-----|
+ 7 PRODJOBA 84| *|-----|-----|-----|-----|-----|-----|-----|-----|
+ 8 XYZJOB 773| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 12 TSOUSERX 310| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 15 TESTJOB1 854| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 19 TSOUSERY 527| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 21 TSOUSERX 880| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 28 TESTJOB1 855| | | | | | | | | | | | | | | | | | | | | | | | | | *<
+ 32 ABCJOB 395| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 35 TSOUSERX 881| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 41 PRODJOBA 85| *|-----|-----|-----|-----|-----|-----|-----|-----|
+ 42 TSOUSERY 528| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 45 TESTJOB1 859| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 49 TSOUSERX 310| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 53 TSOUSERY 881| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 56 TSOUSERY 529| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 61 TSOUSERZ 195| | | | | | | | | | | | | | | | | | | | | | | | | | *<|-----|-----|
+ 62 XYZJOB 773| | | | | | | | | | | | | | | | | | | | | | | | | | >*
+ 63 PRODJOBA 853| | | | | | | | | | | | | | | | | | | | | | | | | | -->*

```

Figure 66. PLOT minor command output

WSIZ nnn

Controls the size of the work area used for the DATA minor command of SEEK

Type:

Minor of SEEK

The variable *nnn* is the size of the work area in kilobytes.

Address space minor commands

The commands this section describes are all minor commands of the address space selection majors commands. They are most commonly used with the JAS*nnn* or JOBN commands, because these select a single address space on the basis of ASID*nnn* or name. You can, of course, use the address space minors with majors such as BATX and TSOJ, where they report on the first job selected.

Use the authorized address space minors only with the address space majors that select one job at a time; this reduces the chance of an address space being accidentally modified when the display line changes unexpectedly.

All of these minors except for TADR and TSNM require the action character (-) in front of them to execute. It specifies that the command execute only once, and changes to a comment character (>) after this execution.

MCTL

Specifies CPU time limit for an address space.

Type:

Minor of address space majors

MCTL is authorized because it circumvents any job class restrictions. You can use MCTL to prevent the abend of a critical job if it takes more CPU time than was specified. MCTL may result in CPU time limits that are rounded up (by one second) from the user specification, due to the conversion between real time and time of day clock units.

Format:

{-}MCTL{=|+|-}{nnnn}{H|M|S}

- An action character.

- =**
Sets the CPU time limit equal to the value specified by the *nnnn* argument.
- +**
Adds the value *nnnn* to the CPU time limit.
- Subtracts the *nnnn* value from the CPU time limit.
- nnnn**
Decimal value to be added/subtracted/set.
- H**
Specifies hours as the unit of time.
- M**
Specifies minutes as the unit of time.
- S**
(Default) Specifies seconds as the unit of time.

For example:

```
>MCTL=23 H
+      >> TIME LIMIT IS NOW 23:00 HR <<
```

MNSW

Marks an address space nonswappable by the SRM.

Type:

Minor of address space majors

OMEGAMON uses a standard SRM interface to mark the current job step nonswappable.

For example, to mark job PAYROLL nonswappable, enter:

```
JOBN  PAYROLL
-MNSW
```

MSWP

Marks an address space swappable by the SRM.

Type:

Minor of address space majors

The SRB routine uses a standard SRM interface to mark the current job step as subject to swapout.

For example, to mark job PAYROLL swappable, enter:

```
JOBN  PAYROLL
-MSWP
```

SWPI

Causes the SRM to swap in an address space.

Type:

Minor of address space majors

To cause the SRM to swap in an address space, OMEGAMON invokes a standard SRM interface that forces a swap-in.

Note: To force the swap-in to occur, the SRM causes the target user to disregard the MPL constraints for the rest of the job step. For example, a job in a domain with the maximum MPL set to zero (that is, a swapout domain) can swap in and accumulate service after you issue SWPI.

For example, to swap in job PAYROLL, enter:

```
JOBN  PAYROLL
-SWPI
```

The SRM swaps users in and out of memory on the basis of dynamically calculated recommendation. Because SWPI does not affect these, the SRM will most likely swap the indicated user out again almost immediately. This limits the usefulness of the SWPI command.

TADR

Displays the unit address for the terminal of a TSO user.

Type:

Minor of address space majors

TADR is authorized because TSO keeps this information in a fetch-protected control block. (Note that VTAM terminals typically have no unit address; use TSNM instead.) Here is an example:

```
TS0J  TS145  TS031  TS089
TADR   345   34F   341
```

TSNM

Displays the symbolic name of a TSO terminal as it is defined to TCAM or VTAM.

Type:

Minor of address space majors

TSNM is authorized because TSO keeps this information in a fetch-protected control block. Here is an example:

```
TS0J  TS145  TS031  TS089
TSNM  L0413  L0416  RCRT0005
```

Program library facilities

ALIB

Displays the authorized program facility (APF) library names and volumes.

Type:

Minor of SYS

```
SYS    >> WLM Goal mode OPT=00 SYSRES=(G1A01C,824A) <<
ALIB   Size of the APF table is 1720 bytes, it contains 75 entries
+     System Libraries
+     MVS999  SYS1.LINKLIB
+     MVS999  SYS1.SVCLIB
+     User Libraries
+     MVS999  MY.DATASET
```

The format of this command is ALIBnn or ALInnn. The two or three digit operand suppresses the display of the first nn or nnn entries in the list. This is useful if all of the data set names do not fit on one screen.

APFU

Updates the Authorized Program Facility (APF) library list.

Type:

Immediate

This command adds or deletes a library, or changes a volume serial name in an existing entry of the APF list.

Format:

```
{-}APFU{A|C|D|L} {DSN=aa.bb.cc} [,VOL=nnnnnn] [,NVOL=nnnnnn]
```

-

An action character. It specifies that the command execute only once, and changes to a comment character (>) after this execution.

A

Adds an entry to the list.

C

Changes an existing entry in the list.

D

Deletes an entry from the list.

L

Lists all of the entries in the APF table.

DSN=

Specifies the name of the data set you want to modify.

VOL=

Specifies the volume serial identification associated with the data set you want to modify.

VOL=*****

Indicates the volser of the SYSRES pack.

VOL=*SMS**

Indicates SMS managed libraries.

NVOL=

Specifies the new volume serial identification, when changing an entry with the C operand.

NVOL=*****

Indicates the volser of the SYSRES pack.

NVOL=*SMS**

Indicates SMS managed libraries.

When any modification is made to the APF list, the Link List Table is also appropriately updated with regard to the APF status of the library in the link list concatenation.

Here is an example of using the C operand:

```
-APFUC DSN=SYS2.NCP.LOADLIB,VOL=SYS005,NVOL=SYS006
```

The following is a typical partial display of APFU with an L operand:

```
>APFUL Size of the APF table is 1720 bytes, it contains 75 entries
+ System Libraries
+ MVS134 SYS1.LINKLIB
+ MVS134 SYS1.SVCLIB
+ User Libraries
+ MVS134 SYS1.VTAMLST
```

LPAM

Lists directory members in the link pack area (LPA).

Type:

Immediate

The LPAM command can list LPA directory information. Use the z/OS SETPROG operator command to add or delete LPA members.

Format:

```
{-} LPAM PGM=cccccccc
```

-

An action character. It specifies that the command execute only once and changes to a comment character (>) after this execution.

PGM=

Specifies the program module name to be processed. OMEGAMON processes this module along with all of its aliases (unless you use the NOALIAS parameter). If you specify a load module name only by an alias, the module is still processed.

The next example shows LPAM being used with no operands. The first line displays the address of the Contents Directory Entry (CDE) from the active LPA queue with the module's entry point and load point addresses from that CDE. If the module is not found on the active LPA queue, this output line does not appear. When a new version of the load module is installed, the active LPA queue reflects the new entry point and load point addresses.

The second line of the display gives the address of the Link Pack Directory entry (LPDE), with the module's entry point and load point addresses from that LPDE. If the module is not in the LPDE, LPAM reports that the module was not found.

```
-LPAM PGM=IGG019AJ  
+   IGG019AJ Is in LPA Directory,   LPDE=00D07558  EP=00E89D60  LP=00E89D60  
+   IGG019AJ has the following alias entry points:  
+   (none)
```

Chapter 12. Degradation Exception Analyzer (DEXAN)

Degradation is unproductive time or the time a workload spends waiting for a system resource, as opposed to actively using that resource. For example, waiting for CPU is degradation because it degrades productivity as opposed to actively using the CPU. Actively using CPU can also degrade productivity if the program is written inefficiently or contains a loop. The Degradation Exception Analyzer (DEXAN for MVS), DEXAN shows where degradation is occurring and how much effect the degradation is having on productivity. This greatly enhances your ability to diagnose and solve performance problems as well as generally increase your system's productivity. DEXAN lets you examine degradation for individual address spaces (or jobs) or the entire system.

The INSPECT feature of DEXAN is useful for identifying this kind of degradation. This feature is explained in [Chapter 14, "Applications tuning," on page 303](#)

To examine degradation caused by workloads contending for resources, see [Chapter 13, "Impact analysis," on page 287](#).

How DEXAN works

DEXAN analyzes degradation by surveying system resources to determine where a workload is spending its time and whether it is spending that time productively or unproductively (whether it is active or waiting). DEXAN statistically analyzes the data it collects when it surveys the system. Thus, DEXAN must collect enough data so that its analysis is statistically valid. Generally, for the analysis to be statistically valid, DEXAN must take at least 30 samples.

User-specified intervals determine the frequency with which DEXAN surveys the system. These intervals can range from 0.1 to 9.9 seconds. The length of the sampling interval affects the amount of CPU overhead associated with the use of DEXAN; CPU overhead increases as the sampling interval decreases. However, even with a sampling interval as low as 0.1 seconds, DEXAN CPU overhead should only be about 1 to 2 percent of total CPU utilization. To monitor DEXAN overhead easily, use the OMEGAMON MFCPU command which allows you to use DEXAN and OMEGAMON commands on the same screen.

For every sample, DEXAN analyzes every address space and determines if and why it is waiting. Typical wait reasons include:

- I/O waits
- Enqueues
- Tape mounts pending
- Disk mounts pending
- Waiting for CPU cycles
- Swapping
- Paging
- Waiting for JES service
- Waiting for HSM service

By sampling at a regular interval, DEXAN can statistically estimate the relative impact each wait reason has on system performance. For example, assume that TSO response time is two seconds and TSO waits are distributed as follows:

- Using CPU 10%
- Paging 50%
- Unilateral swaps 35%
- Device 3AQ 5%

Using CPU does not constitute degradation and accounts for 10% of TSO response time. Degradation accounts for the remaining 90% of TSO response time. Statistically, you can estimate the impact of the three types of degradation as follows:

- Paging = 50% of 2.0 seconds = 1.0 second
- Unilateral swaps = 35% of 2.0 seconds = 0.7 seconds
- Device 3AQ busy = 5% of 2.0 seconds = 0.1 seconds

This indicates where to tune, as well as the potential benefit of tuning.

When a system I/O reconfiguration occurs, DEXAN users who are operating on a system with devices defined as dynamic will receive a series of informational messages stating that the collector is being recycled, and the DEXAN subtask has been stopped and started again. This is shown in the following example:

```
DEX
+ >> DX1070 I/O CONFIGURATION CHANGE DETECTED <<
+ >> DX1071 COLLECTOR BEING RESTARTED DUE TO POSSIBLE I/O CONFIGURATION CH
+ >> DX1000 THE DATA COLLECTOR STARTED. WORKAREA SIZE= 97300 BYTES <<
+ COLLECTOR MONITORING DEVICES DEFINED AS DYNAMIC
+ COLLECTOR NOT MONITORING LATEST I/O CONFIGURATION
```

How to use DEXAN commands

DEXAN commands can only be used if the DEXAN collector has been initiated. Like OMEGAMON commands, DEXAN commands are entered in column 2.

Some DEXAN commands are minor commands of the DEX major command. Therefore, DEX must sometimes be present on the screen for other DEXAN commands to work. For example, to start the data collector you would enter DEX and BEGN in the following way:

```
DEX
BEGN
```

DEXAN display commands

There are two commands, DEX and IPRO, that you can use to display information regardless of whether you are analyzing degradation for individual address spaces or the entire system. Other DEXAN display commands show degradation data for a specific area, and are described along with the family of commands that you use to analyze degradation for that area.

DEX

Displays information about the status of the data collector.

Type:

Major

The DEX command must precede any DEX minor commands on the screen for them to work.

IPRO

Displays resource impact profile.

Type:

Minor of DEX

Use the IPRO command preceding a plot command such as IPRO MONJ01 to display a resource impact profile. The plot commands that you may use with IPRO are:

Table 12. Commands used with IPRO	
Command	Explanation
MONJnn	Displays a resource impact profile for the address space with DEXAN slot number nn, jobname, or ASID.
PLTJnn	Displays a resource impact profile for the address space with DEXAN slot number nn, jobname, or ASID.
PLTAnn	Displays a resource impact profile for the address space with DEXAN slot number nn.
PLTS	Displays a resource impact profile for system-wide address spaces.

For example, IPRO MONJ01 would display a resource impact profile for the address space with DEXAN slot number 01.

```

DEX    >> DX0000 V420 running. Cycles=149 STIM=2.2 Elap=10:33 MN <<
begn   >> DX2000 Command valid only when Collect not running <<
monj01 ADDR01 ASID=281 % |0 1 2 3 4 5 6 7 8 9 0|

```

A typical resource impact profile screen for an address space follows.

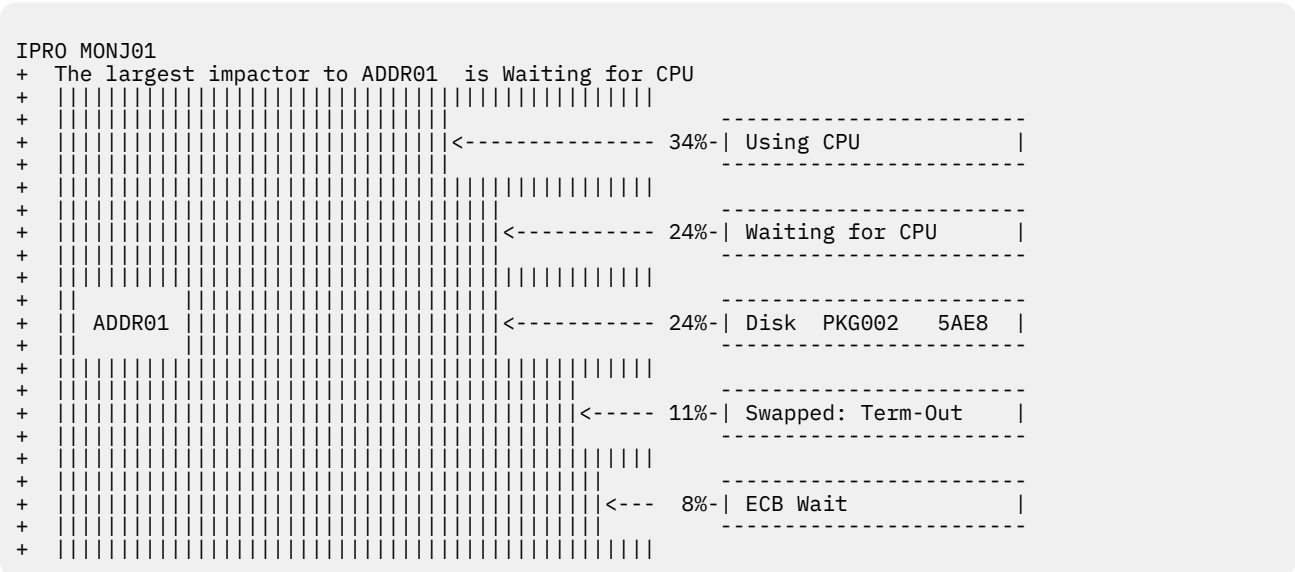


Figure 67. Typical resource impact profile for address space

The resource impact profile is an alternative to the plot display for showing degradation data. The resource impact profile shows more graphically the impact of system resources on a workload. The vertical lines represent a workload’s productivity. For wait reasons the arrows intruding from the right represent the impact a resource has on productivity. The greater the indentation, the greater the impact the system resource has on productivity. The indentations correlate directly to the actual percentage of time the workload waited for or used a resource.

Because the resource impact profile is such a telling graphic display, it lends itself to quick interpretation. This makes it particularly helpful in situations that require fast response from you.

There may be more execution states than can be displayed on the resource impact profile at one time. To see all of the execution states on the display, scroll forward. Execution states are always listed from greatest impact to least impact except for Using CPU, which is always listed first. In general, all of the execution states fit on one screen.

Controlling the DEXAN data collector

The commands described in this section are used to start and stop the data collector and to control aspects of its performance, such as resetting counters, setting analysis sampling intervals, and synchronizing the CLR interval with the RMF interval.

BEGN

Starts DEXAN data collector.

Type:

Minor of DEX

To start the DEXAN data collector, use the BEGN command. For example:

```
DEX
BEGN
```

results in:

```
DEX    >> DX0001 Collector has not been started. <<
>BEGN  >> DX1000 The data collector started.  Workarea size=    38696 bytes. <<
```

Note that the command is automatically changed to a comment (greater than (>) sign in column 1) so that the command is not executed again.

The BEGN command also places an asterisk (*) on the INFO-line between the product and version numbers. This asterisk remains until you issue the END minor command.

In ISPF mode, APF authorization is required to issue this command.

CENQ

Controls enqueue analysis.

Type:

Minor of DEX

CENQ without an operand displays current enqueue wait collection setting. CENQOF turns off enqueue wait collection. CENQON turns on enqueue wait collection. CENQ nn turns on enqueue wait collection and causes enqueue sampling to take place every nn cycles (valid range is 1 - 10).

Note: By default, the enqueue analysis is turned off to conserve CPU resources, which may amount to as much as 5%. When it is turned off, enqueue waits are attributed to ECB waits. Thus, the ECB percentages increase.

CLR nnn

Resets the data collector counters every nnn minutes.

Type:

Minor of DEX

To clear the DEXAN counters while the data collector is running, use the CLR nnn command (with SYNCOF), where nnn is the interval in minutes past the hour at which DEXAN is to clear its counters. For example, if you use a numeric argument of 15, the counters are cleared immediately, and then every 15 minutes starting 15 minutes after the hour.

If nnn is greater than 60, the counters are cleared immediately, and then every nnn minutes after.

By default, the counters are cleared every 30 minutes. If nnn is omitted, the counters are cleared immediately and the current clear interval is displayed. For example:

```
DEX    >> DX0000 V420 running. Cycles=214 STIM=1.0 Elap= 1:30 MN    <<
>CLR   >> DX7000 Collection counter clearing interval is every 30 minutes <<
```

DEXAN does the following:

1. Clears all DEXAN counters.
2. Sets the number of cycles to zero.
3. Sets the DEXAN elapsed time to zero.

This is logically equal to stopping and then restarting DEXAN. Note that the command is changed into a comment so that it is not re-executed on the next cycle. The CLR command can only be used after the data collector has been started (by default, the interval is set to 30 minutes).

Note: After an episode of severe degradation, such as a ten minute tape mount pending, you may wish to clear the DEXAN counters to prevent this from overshadowing or masking other significant sources of degradation.

END

Stops the data collector.

Type:

Minor of DEX

To stop the DEXAN data collector, use the END command. For example:

```
DEX
END
```

DEXAN acknowledges your request and stops collecting data. NUMAnn, NUMFnn and NUMPnn can now be changed. However, to restart DEXAN you must now use BEGN.

Note: Stopping OMEGAMON automatically stops DEXAN including the data collector. Terminating DEXAN does not terminate OMEGAMON.

RESM

Restarts the data collector.

Type:

Minor of DEX

When you want to restart the DEXAN data collector, use the RESM command. For example:

```
DEX >> DX9004 Collector suspended since 10:54:56 Cycles=619 Elap 25:03 MN <<
RESM >> DX0003 Collector being resumed <<
```

STIMnn

Sets the sampling interval for data collection.

Type:

Minor of DEX

To change the DEXAN data collector sampling time, use the STIMnn command, where nn is the sampling time in tenths of a second. If you omit the nn argument, DEXAN displays the current value of STIM. For example:

```
DEX
STIM
```

results in:

```
DEX >> DX0001 Collector has not been started. <<
>STIM >> DX7100 The data collector sampling time = 2.2 seconds. <<
```

The default sampling time is 2.2 seconds. To change this value, enter STIM with a numeric operand. For example:

```
DEX
STIM04
```

results in:

```
DEX    >> DX0001 Collector has not been started. <<
>STIM04 >> DX7100 The data collector sampling time = .4 seconds. <<
```

Note: Changing STIM while DEXAN is active is permitted. However, statistical integrity of the data is compromised until the end of the clear interval. Therefore, be careful about interpreting results if you change the STIM value while the data collector is running.

In TSO mode on a non-SE/SP system, you should set the sampling time to 0.4 seconds or less to prevent the DEXAN TSO transaction from being swapped out after each STIMER. As an alternative, you can use OMEGAMON to make the TSO ID non-swappable.

SUSP

Suspends the data collector.

Type:

Minor of DEX

If you want to freeze data collection so that DEXAN analyzes several workloads over the same period of time, you can suspend data collection temporarily without losing the collected data. To do this, use the SUSP command. For example:

```
DEX    >> DX0000 V420 running. Cycles=605 STIM=2.2 Elap=22:48 MN <<
SUSP   >> DX0002 Collector being suspended <<
```

DEXAN acknowledges your entry and suspends data collection. All display commands such as PLTnnnn remain available.

SYNC

Synchronizes the CLR interval with the RMF interval.

Type:

Minor of DEX

The SYNCON command synchronizes the DEXAN CLR interval with the RMF interval, if RMF is running. The SYNCOFF command allows DEXAN data collection to run independently from RMF.

With SYNCON, DEXAN output covers the same time period as RMF, so display figures are comparable to RMF output. The default is SYNCOFF, with a clear interval of 30 minutes. When SYNCON is specified, the CLR command is ignored until a SYNCOF command is issued.

Note: The SYNCOFF command restores the clear interval to 30 minutes, even if you used a CLR command previously to change the clear interval.

You can use SYNC as a shortened version of SYNCON.

Job degradation analysis

DEXAN job degradation analysis lets you look at degradation for selected jobs (address spaces).

Overview of job analysis data collection

Unlike system-wide analysis, job analysis does not automatically start when DEXAN data collection starts. Each job that is individually analyzed requires additional main storage and additional processing resources. For this reason, DEXAN has commands to control which jobs are analyzed.

The following commands control job analysis:

ADEL

Turns off monitoring for a specified job.

ALST

Lists jobs that are currently being monitored.

ASEL

Selects a job for analysis and begins monitoring that job.

MONJ

Selects a job for analysis and begins monitoring that job. Subsequent execution of MONJ for that job produces a plot display of the wait reasons. Essentially, MONJ combines the ASEL and PLTJ commands.

NUMA

Sets the maximum number of address spaces that DEXAN can monitor at one time. The default is 3.

For more information about the use of these commands, see [“Job analysis control commands” on page 273](#).

Overview of job analysis data display

Job analysis data can be displayed in several formats.

Resource Impact Profile

A high level display of job degradation that lends itself to quick interpretation. It reveals the impact system resources are having on an individual job. Use the IPRO command with either the MONJ, PLTA, or PLTJ command to display a resource impact profile. To see a typical resource impact profile, refer to [Figure 68 on page 281](#).

Plot

A more detailed display that combines numeric data with graphics. Use either the PLTA, PLTJ, or MONJ command to display a plot.

Percentage

A listing of the top wait reasons by percentage without any graphic display of that data. Use either the PCTA or PCTJ command to list wait reasons by percentage.

Count

A listing of actual number counts from which the other three displays are based. Use the CNTA or CNTJ command to list a count of wait reasons.

Job analysis control commands

You can use either the ASEL or MONJ command to select a job for monitoring. DEXAN assigns a DEXAN slot number to each job selected with either ASEL or MONJ. The first job selected gets number 01, the second 02, and so on. Once a DEXAN job number is assigned, the ADEL, PLTJ, PCTJ, and MONJ commands let you enter either the DEXAN job number or the job name as an operand, for example, MONJ JES2 or MONJ3.

ADELnn

Removes job *nn*, jobname, or ASID from analysis.

Type:

Minor of DEX

Format

ADELnnADEL jobnameADEL ASID=asid

To delete a job (or address space) from job analysis, use the ADEL command followed immediately the two-digit DEXAN slot number, by a space and the jobname, or by a space and the ASID. For example:

```
DEX >> DX0000 V420 running. Cycles=99 STIM=2.2 Elap=3.46MN <<
>BEGN >> DX1000 The Data Collector Started. Workarea size= 38512 Bytes <<
ADEL01
ADEL TCAM
```

Here, use of both slot number and jobname are shown, ADEL01 and ADEL TCAM.

```
DEX >> DX0000 V420 running. Cycles=25 STIM=2.2 Elap=3.46MN <<
>BEGN >> DX1000 The Data Collector Started. Workarea size= 38512 Bytes <<
ADEL01 JOB1 ASID=300 >> DX1050 Entry will be deleted. <<
ADEL02 TCAM ASID=325 >> DX1050 Entry will be deleted. <<
```

These two address spaces are no longer included in job analysis.

ALST

Displays jobs being analyzed by DEXAN.

Type:

Minor of DEX

To display jobs included in DEXAN job analysis, use the ALST command. DEXAN displays all jobs currently included in the analysis. For example:

```
DEX > V420 running. Cycles=25 STIM=2.2 Elap=3.46MN <<
>BEGN >> The Data Collector Started. Workarea size= 38512 Bytes <<
ALST
```

ALST displays monitored jobs with their DEXAN slot number and ASID.

```
DEX >> DX0000 V420 running. Cycles=99 STIM=2.2 Elap=3.46MN <<
>BEGN >> DX1000 The Data Collector Started. Workarea size= 38512 Bytes <<
ALST01 JOB1 ASID=300 >> DX1600 Collection started 7:03:04, elap 1:24 MN, items=39 <<
+ 02 TCAM ASID=325 >> DX1600 Collection started 7:04:02, ended 2:34 MN, items=53 <<
```

Note from the previous figure:

- The time when monitoring started.
- Either the elapsed time of the monitoring or the time when the monitoring ended. This means that the job completed while being monitored, not that you ended monitoring. Note that, if ADEL was used to stop monitoring, that job would not be displayed.
- The number of samples collected (items) shows whether statistical significance has been reached. For statistical significance, this number should be at least 30.

This example shows that two address spaces have been selected for job analysis. Of these, TCAM has already terminated, so only JOB1 is still running and being monitored.

ASEL

Selects a job for analysis and begins monitoring that job.

Type:

Minor of DEX

- Without a numeric argument, NUMA displays the maximum number of address spaces that you can monitor.
- With a numeric argument, NUMA changes the maximum number of address spaces that you can monitor. By default, DEXAN allows you to monitor up to five address spaces.

To display the maximum number of address spaces, use the NUMA command without an operand. For example:

```
DEX    >> DX0001 Collector has not been started. <<
NUMA   >> DX1100 DEXAN to support up to 5 address space analyses. <<
```

For example:

```
DEX    >> DX0001 Collector has not been started. <<
>NUMA6 >> DX1100 DEXAN to support up to 6 address space analyses. <<
```

The valid range of the numeric argument is 0 through 99. An entry of zero turns off support for individually monitored address spaces. High values for the numeric argument can increase the storage overhead. The maximum number of individually monitored address spaces cannot be changed after the DEXAN data collector has been started.

Job degradation display commands

Use these commands to display degradation data for address spaces.

MONJnn

Displays plot for DEXAN slot number *nn* or jobname (after first execution).

Type:

Minor of DEX

Executing MONJ a second time results in the following plot:

```
MONJ01 JOB1
+ Using CPU          5.6 |--> . . . . . . . . . .|
+ Waiting for CPU   33.0 |-----> . . . . . . . . . .|
+ Private Page-in Wait 20.4 |-----> . . . . . . . . . .|
+ Control Unit      15.0 |-----> . . . . . . . . . .|
+ Disk MVS307      14.9 |-----> . . . . . . . . . .|
```

Note: You may have noticed that the individual wait states shown in this display do not add up to 100%. By default, DEXAN excludes from the display wait reasons that account for less than 5% of the total wait time. You can use the THRS command to adjust this threshold. The MONJ command is also described under “Job analysis control commands” on page 273, because this command can control data collection and display data.

PLTAnn

Displays a plot of job degradation analysis data for DEXAN slot number *nn*.

Type:

Minor of DEX

PLTJnn

Displays a plot of job degradation analysis data for DEXAN slot number *nn*, jobname, or ASID.

Type:

Minor of DEX

To display a plot of the most significant wait reasons affecting a job, use the PLTA, PLTJ, or MONJ command.

A sample use of the MONJ command appears earlier in this section. Following is an example of the PLTA and PLTJ command. For example:

```

PLTJ02
=====
PLTJ T010HP32
=====
PLTA05

```

In this example, both the DEXAN slot number and jobname were used.

```

PLTJ02 T063D ASID=279      % |0  1  2  3  4  5  6  7  8  9  0|
+ Using CPU                5.0 |--> .  .  .  .  .  .  .  .  .  .  .|
+ Disk TS0001  253 Queued  30.0 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Private Page-in Wait    20.0 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Control Unit  25X Queued  15.0 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Disk MVS307   735 Active  15.0 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Waiting for CPU         5.0 |--> .  .  .  .  .  .  .  .  .  .  .|
=====
PLTJ03 T010HP32 ASID=332  % |0  1  2  3  4  5  6  7  8  9  0|
+ Using CPU                24.5 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Disk MVWK56  137 Queued  62.1 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Waiting for CPU         5.2 |--> .  .  .  .  .  .  .  .  .  .  .|
=====
PLTA05 SCJ05FL ASID=472   % |0  1  2  3  4  5  6  7  8  9  0|
+ Using CPU                5.1 |--> .  .  .  .  .  .  .  .  .  .  .|
+ Tape Mount Pending      48.8 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Swapped: Det-Wait      25.4 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Logical Channel 14 Queued  7.2 |--> .  .  .  .  .  .  .  .  .  .  .|
+ Tape 010259   A99 Active  6.4 |--> .  .  .  .  .  .  .  .  .  .  .|

```

Using CPU, which is a productive state, always appears first on a plot.

In this example, both job T063D and T010HP32 suffer from severe degradation because of disk I/O contention. Job SCJ05FL waits for almost half of its time on Tape Mounts Pending.

PCTAnn

Displays wait reasons by percentage for job *nn*.

Type:

Minor of DEX

Use the PCTA command with the DEXAN slot number to display a percentage listing of wait reasons.

PCTJnn

Displays wait reasons by percentage for job *nn* or jobname.

Type:

Minor of DEX

Use PCTJ command to display the wait reasons for a job (address space).

For example:

```

PCTJ OMEGAMON
PCTA5

```

results in:

```

PCTJ02      CPU STI
+ OMEGAMON  0.4 99.6
PCTA05      CPU PAG  UNI  CPW  4A3  4A0  A9A  4A4  157 Q4A4  140
+ PRMS40BW  9.8 28.2 21.7 18.4  9.0  7.9  5.8  4.6  2.1  1.9  .6

```

This display shows two address spaces: OMEGAMON and PRMS40BW. OMEGAMON shows only STI waits, but PRMS40BW shows degradation due to paging, unilateral swaps, waiting for CPU cycles, and I/O contention.

The first row of the percentage display lists execution states where a workload spends time. The numbers in the second row are the percentage of time that a workload spends in an address space.

For a listing of all possible execution states, see [Figure 68 on page 281](#). For more information on execution states, see [“Execution state summary” on page 282](#).

Only 14 execution states can be displayed at one time. To scroll horizontally to see any additional wait reasons, type 1 in column 1 next to the command to move to the second screen. Type 2 in column 1 to move to the third screen, and so on. To return to the first screen, enter 0 or blank in column 1. Likewise, type 1 in column 1 next to the command to return to the second screen from any screen.

CNTAnn

Displays counts of wait reasons for DEXAN slot number *nn*.

Type:

Minor of DEX

CNTJnn

Displays counts of wait reasons for DEXAN slot number *nn* jobname, or ASID.

Type:

Minor of DEX

For example:

```
CNTJ USERPSP
```

results in:

```
CNTJ01    CPU   PAG   UNI   CPW   4A3   4A0   A9A   4A4   QA9A   157   Q4A0   140
+ USERPSP  ---   307   236   206   98    86    71    50    35    23    21    10
```

The counts show you the actual results of DEXAN sampling or data collection. Each column heading is an execution state where a job is spending time. The numbers are the actual number of times DEXAN found a job in a particular execution state.

No counts appear for CPU because using CPU is determined by an alternate method. For a complete list of all possible execution states, see [Figure 68 on page 281](#). For more information about execution states, see [“Execution state summary” on page 282](#). These counts are the raw data from which the resource impact profile, plot, and percentage are drawn. These three displays are what you will rely on most of the time because they are easier to interpret than a count. However, it is good to become familiar with counts because they can help you become familiar with your system and with DEXAN. Also, if you know where the displays come from, they will be even easier to use.

The CNTJ command cannot be used with the IPRO command.

System-wide analysis

DEXAN lets you analyze degradation on a system-wide basis. System-wide degradation analysis means monitoring all address spaces combined, and reporting on them as one entity. This analysis lets you look at the overall impact a system resource is having on the productivity for all address spaces.

System-wide monitoring begins when you start DEXAN data collection with the BEGN command. The only commands that accompany system-wide analysis are display commands.

Displaying system-wide degradation data

You can display degradation data for the system in four formats.

Resource Impact Profile

A high level display of system degradation that lends itself to quick interpretation. It shows a picture of the impact system resources are having on workloads running on the system. Use the IPRO command with the PLTS command to display a resource impact profile for system-wide analysis. To see a typical resource impact profile, refer to [Figure 67 on page 269](#).

Plot

A more detailed display that combines numeric data with graphics. Use the PLTS command to display a plot of system-wide analysis.

Percentage

A listing of wait reasons by percentage without any graphic display of that data. Use the PCTS command to display a percentage listing of wait reasons for system-wide analysis.

Count

A listing of actual number counts from which the other three displays are based. Use the CNTS command to list counts of wait reasons.

PLTS

Plots wait reasons system-wide.

Type:

Minor of DEX

To display a plot of the most significant wait reasons system-wide, use the PLTS command. For example:

```
PLTS          (136.0 jobs)    % |0  1  2  3  4  5  6  7  8  9  0|
+ Using CPU          .8 |>  .  .  .  .  .  .  .  .  .  .  .|
+ Waiting for CPU    42.3 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ ECB Wait          21.2 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ STIMER Wait       12.6 |-----> .  .  .  .  .  .  .  .  .  .  .|
+ Swapped: Detected Wt 10.4 |----> .  .  .  .  .  .  .  .  .  .  .|
+ ECB (w/STIMER)     5.7 |--> .  .  .  .  .  .  .  .  .  .  .|
```

PCTS

Displays a percentage listing of wait reasons system-wide.

Type:

Minor of DEX

Use the PCTS command to display execution states. For example:

The first row in the percentage display shows execution states where a workload is spending time. The numbers are the percent of time that the workload is spending in an execution state. For a listing of all execution states, see [Figure 68 on page 281](#). For more information about execution states, refer to [“Execution state summary” on page 282](#).

In this example, the major bottleneck is the CPU. The system is waiting almost half the time for CPU cycles.

CNTS

Displays counts of wait reasons system-wide.

Type:

Minor of DEX

To display counts of wait reasons for the system, use the CNTS command. For example, the command CNTS results in:

```
CNTS          CPU  ECS  ECB  STI  DET  CPW  LON  TMP  TOU  SWI  333  155
+            ---  5913  2967  1760  1451  802  250  175  161  46  30  29
```

The counts show you the actual results of DEXAN sampling or data collection. Each column heading is an execution state where system transactions are spending time. These numbers are the number of times DEXAN found a workload in a particular execution state. For a complete list of all possible execution states, see [Figure 68 on page 281](#). For more information about execution states, see [“Execution state summary” on page 282](#).

No counts appear for CPU because using CPU is determined by an alternate method.

These counts are the raw data from which the resource impact profile, plot, and percentage are drawn. These three displays are what you will rely on most of the time because they are easier to interpret.

Controlling the DEXAN data display

The commands for controlling DEXAN data display are summarized in the following table.

Command	Description	Default
BLST	Displays all collection buckets and their status	N/A
BOF	Removes a collection bucket from the display	N/A
BON	Adds a collection bucket to the display	N/A
THRS	Limits plot output to the most important wait reasons (by percent of total wait time)	5%

BLST

Lists collection buckets for display.

Type:

Minor of DEX

You can list the collection buckets that are currently included in the DEXAN displays with the BLST command. The names correspond to the code names that describe wait reasons. See [“Execution state summary” on page 282](#) for an explanation of the wait reason code names. For example:

```
DEX    >> DX0001 Collector has not been started <<
BLST  COM=on  CPW=on  DMP=on  ECB=on  ECS=on  PSO=on  CAN=on  JST=on
+     DEL=on  RQU=on  RCL=on  RCV=on  MIG=on  BKP=on  RCR=on  HLS=on
+     CIL=on  DLD=on  LCK=on  MSS=on  PAG=on  RDY=on  RT0=on  STI=on
+     SWI=on  TMP=on  WTO=on  TOU=on  TIN=off LON=on  AUX=on  RST=on
+     DET=on  REQ=on  EEX=on  EXC=on  UNI=on  TSW=on  CSS=on  SPS=on
+     OLS=on  APS=on
```

Note: The default is for all wait reasons to be displayed except for terminal input waits (TIN).

You can list the names and descriptions of all collection buckets by placing the slash (/) help symbol in the label field (column 1) before the BLST command. DEXAN displays the names and descriptions of each bucket as comment lines. For example:

```
DEX    >> DX0001 Collector has not been started <<
/BLST
```

results in the following display.

```

DEX    >> DX0001 Collector has not been started <<
>blst >> DEXAN -- List bucket items selected for display. <<
>      COM -- LPA or CSA page in wait.
>      CPW -- CPU-wait.
>      DMP -- Disk mount pending.
>      ECB -- ECB-wait.
>      ECS -- ECB-wait with STIMER.
>      PSO -- JES PROCESS SYSOUT request queued.
>      CAN -- JES job CANCEL request queued.
>      JST -- JES job STATUS request queued.
>      DEL -- JES job DELETE request queued.
>      RQU -- JES job REQUEUE request queued.
>      RCL -- HSM RECALL dataset request.
>      RCV -- HSM RECOVER dataset request.
>      MIG -- HSM MIGRATE dataset request.
>      BKP -- HSM BACKUP dataset request.
>      RCR -- HSM read control record request.
>      HLS -- HSM TSO HLIST request.
>      CIL -- HSM read JES3 C/I request.
>      DLD -- HSM DELETE dataset request.
>      LCK -- Local or global lock wait.
>      MSS -- Waiting for staging.
>      PAG -- Local page-wait.
>      RDY -- User ready to swap in.
>      RTO -- RTO wait.
>      STI -- STIMER-wait.
>      SWI -- Swap in scheduled.
>      TMP -- Tape mount pending.
>      WTO -- Detected wait with WTOR.
>      TOU -- Terminal output wait swap.           1 (MVS swap code)
>      TIN -- Terminal input wait swap.           2
>      LON -- Long wait swap.                     3
>      AUX -- Auxiliary storage swap.             4
>      RST -- Real storage swap.                 5
>      DET -- Detected wait swap.                6
>      REQ -- Request swap.                      7
>      EEX -- Enqueue exchange swap.            8
>      EXC -- Exchange swap.                    9
>      UNI -- Unilateral swap.                  10
>      TSW -- Transition swap.                  11
>      CSS -- Central storage swap.             12
>      SPS -- System paging swap.               13
>      OLS -- Swapped out too long.             14
>      APS -- APPC wait swap.                   15

```

Figure 68. Collection bucket descriptions (/BLST)

BOFaaa

Removes collection bucket aaa from display.

Type:

Minor of DEX

BOFa^{aaa} removes the aaa collection bucket from the display. The bucket is still collected, but the results are not included in the display. For example:

```

DEX
BOFDMP

```

results in:

```

DEX    >> DX0001 Collector has not been started <<
BOFDMP >> DX3002 The collection counter for DMP is now off <<
BLST  COM=on CPW=on DMP=off ECB=on ECS=on PSO=on CAN=on JST=on
+     DEL=on RQU=on RCL=on RCV=on MIG=on BKP=on RCR=on HLS=on
+     CIL=on DLD=on LCK=on MSS=on PAG=on RDY=on RTO=on STI=on
+     SWI=on TMP=on WTO=on TOU=on TIN=off LON=on AUX=on RST=on
+     DET=on REQ=on EEX=on EXC=on UNI=on TSW=on

```

BONaaa

Selects collection bucket aaa for display.

Type:

Minor of DEX

BONaaa adds the aaa collection bucket to the display. For example:

```
DEX    >> Collector has not been started  <<
BONDMP >> Done <<      Disk Mount Pending
```

results in:

```
DEX    >> DX0001 Collector has not been started  <<
BONDMP >> DX3001 The collection counter for DMP is now on <<
BLST   COM=on CPW=on DMP=on ECB=on ECS=on PS0=on CAN=on JST=on
+      DEL=on RQU=on RCL=on RCV=on MIG=on BKP=on RCR=on HLS=on
+      CIL=on DLD=on LCK=on MSS=on PAG=on RDY=on RT0=on STI=on
+      SWI=on TMP=on WTO=on TOU=on TIN=off LON=on AUX=on RST=on
+      DET=on REQ=on EEX=on EXC=on UNI=on TSW=on
```

The DMP collection bucket now returns to the display.

THRS

Limits plot output to the most important wait reasons by setting threshold.

Type:

Minor of DEX

The THRS command limits plot output to the most important wait reasons. It is set by default to 5%. This means that only those wait reasons that account for more than 5% of the total wait time are included in the display. You can change this value by entering an operand with the THRS command. If no operand is specified, THRS displays the current value. For example:

```
DEX
THRS
```

results in:

```
DEX    >> DX0000 V420 running. Cycles=214 STIM=1.0 Elap= 1:30 MN SYNC <<
THRS   >> DX7200 Plot percentage threshold is 5%. <<
```

To change from the default of 5%, enter THRS with an operand. For example:

```
DEX
THRS3
```

results in:

```
DEX    >> DX0000 V420 running. Cycles=229 STIM=1.0 Elap= 1:45 MN SYNC <<
THRS3  >> DX7200 Plot percentage threshold is 3%. <<
```

Note: The screen format allows you to display a maximum of 14 wait reasons.

Execution state summary

The DEXAN wait reasons fall into seven major categories.

- Using CPU
- Swap reason waits
- Miscellaneous waits

- Logical channel waits
- Control unit
- Device waits
- Enqueues

Each of these categories is discussed in turn. The codes shown are for the PCT displays.

Using CPU

Using CPU for a job is always shown as the first item in the display.

Swap reason waits

Table 14 on page 283 lists the SRM swap reasons.

<i>Table 14. List of SRM swap reasons</i>	
Reason	Code
Term-Out	TOU
Term-In	TIN
Long-Wat	LON
Aux-Strg	AUX
Real-Str	RST
Det-Wait	DET
Request	REQ
Enq-Xchg	EEX
Exchange	EXC
Unilatr	UNI
Transwap	TSW
Cent-Str	CSS
Sys-Page	SPS
Too-Long	OLS
APPC-Wt	APS

Advanced Program-to-Program Communication (APPC) enables applications to use SNA LU6.2 to communicate between transaction programs (TPs) running in an SNA environment, either on the same system or on different systems. APPC transaction programs can use z/OS services, such as dataspaces, in their conversations.

Miscellaneous waits

Table 15 on page 283 describes the miscellaneous wait reasons.

<i>Table 15. List of miscellaneous wait reasons</i>		
Reason	Description	Code
HSM Backup Dataset	The address space is waiting for HSM to execute a data set backup request.	BKP
JES Job Cancel	The address space has issued a cancel request to JES2.	CAN
HSM Read JES3 C/I	The address space is waiting for HSM to execute a C/I locate request.	CIL

<i>Table 15. List of miscellaneous wait reasons (continued)</i>		
Reason	Description	Code
Common Page-In Wait	The address space is waiting for a PLPA or Common page-in.	COM
Waiting for CPU	The address space is waiting on the active CPU dispatching queue. High percentages here may mean the address space is low priority, or perhaps a shortage of CPU cycles.	CPW
JES Delete Request	The address space has issued a job delete request to JES2.	DEL
HSM Delete Dataset	The address space is waiting for HSM to delete a data set.	DLD
Disk Mount Pending	The address space is waiting for a disk to be mounted by the operator.	DMP
ECB Wait	The address space has issued a voluntary wait. Examples are an IMS message region waiting for work, a CICS region waiting for work, or OMEGAMON waiting between screen refreshes. High percentages of ECB waits may indicate a program error or normal voluntary waiting as previously discussed.	ECB
ECB Wait with STIMER	The address space has issued a voluntary wait. Examples are an IMS message region waiting for work, a CICS region waiting for work, or OMEGAMON waiting between screen refreshes. An STIMER is outstanding in the address space along with the voluntary wait.	ECS
HSM TSO HLIST	The address space is waiting for HSM to execute a TSO HLIST command.	HLS
JES Status Request	The address space has issued a job status request to JES2.	JST
Waiting for MVS Lock	The address space is waiting to acquire a local or global z/OS lock.	LCK
HSM Migrate Dataset	The address space is waiting for HSM to migrate a data set.	MIG
Waiting for Staging	The address space is waiting for a mass storage volume to be staged.	MSS
Private Page-In Wait	The address space is waiting for a page-in operation.	PAG
JES PROCESS SYSOUT	The address space issued a process SYSOUT request to JES2.	PSO
HSM Recall Dataset	The address space is waiting for HSM to recall a data set.	RCL
HSM Read Control Record	The address space is waiting for HSM to read a control data set record.	RCR
HSM Recover Dataset	The address space is waiting for HSM to recover a backup data set.	RCV

<i>Table 15. List of miscellaneous wait reasons (continued)</i>		
Reason	Description	Code
SRM Delay (MPL)	The address space is swapped out and is ready to be swapped back in, but the SRM has not yet permitted the swap-in. This can occur when a domain is at the target MPL and the SRM has to either increase the target or swap out another address space in the domain. When SRM has scheduled a swap-in via ASM, the work shows up in SWAP PAGE-IN WAIT or in SWI state. .)	RDY
JES2 Requeue Request	The address space has issued a job requeue request to JES2.	RQE
SRM Delay (RTO)	SRM delayed the TSO transaction due to a RTO specification in the IPS.	RTO
STIMER Wait	The address space issued a STIMER and is voluntarily waiting for it to end. This is typically not considered degradation because it is a voluntary wait.	STI
Swap Page-In Wait	This is an ASM condition. The address space has been given to the ASM queue and is waiting for z/OS to swap it in to storage. An excessive value for this wait could be caused by a slow I/O device or DASD contention.	SWI
Tape Mount Pending	The address space is waiting for a tape to be mounted by the operator.	TMP
Swapped with outstanding WTOR	The address space has been swapped out due to Detected Wait, and the address space also has an outstanding WTOR.	WTO

I/O device waits

Table 16 on page 285 describes the control unit and device waits:

<i>Table 16. List of device wait reasons</i>	
Reason	Code
DISK volser <i>aaa</i> ACTIVE	<i>aaa</i>
TAPE volser <i>aaa</i> ACTIVE	<i>aaa</i>
CONTROL UNIT <i>aaX</i> QUEUED	Q <i>aaX</i>
DISK volser <i>aaa</i> QUEUED	Q <i>aaa</i>
TAPE volser <i>aaa</i> QUEUED	Q <i>aaa</i>
DISK volser <i>aaa</i> RESERVED	R <i>aaa</i>

where *aaa* is the address of the device.

Active

An I/O is in progress to the device.

Queued

An I/O is queued waiting for the device to become available for use by this address space.

Reserved

An I/O is waiting because of a reserve or active I/O to this device from another CPU, or because of head-of-string contention from this CPU or another.

Logical Channel Waits apply only to z/OS systems.

Logical channel waits are abbreviated *Lnn*. The long form is LOGICAL CHANNEL *nn* QUEUED.

Enqueues

Table 17 on page 286 lists the enqueues.

Reason	Code
SYSDSN enqueue	*DSN
SYSIEA01 enqueue	*IEA
SYSIEFSD enqueue	*IEF
SYSIGGV1 enqueue	*IG1
SYSIGGV2 enqueue	*IG2
SYSIKJBC enqueue	*IKJ
SYSSMF01 enqueue	*SMF
SYSIEWLP enqueue	*IEW
SYSVSAM enqueue	*VSA
SYSVTOC enqueue	*VTO
SYSZVARY enqueue	*ZVA
Misc SYS enqueue	*SYS
Misc USR enqueue	*USR

For definitions of these enqueues, see Volume 1 of the *IBM MVS/Extended Architecture Debugging Handbook*.

The CENQ command controls enqueue analysis.

Note: By default, the enqueue analysis is turned off to conserve CPU resources, which may amount to as much as 5%. When it is turned off, enqueue waits are attributed to ECB waits. Thus, the ECB percentages increase.

Using the CENQ*nn* format you can reduce the overhead of enqueue analysis and still capture data. Note that the number of cycles required for a significant count is also increased (for example, 40 samples take 200 cycles with CENQ05). The DEXAN counts are also adjusted to account for the delayed sampling.

Chapter 13. Impact analysis

OMEGAMON impact analysis feature is a diagnostic tool that helps to solve problems caused by workload contention.

When you use the impact analysis feature to monitor a workload, you receive the following information about the monitored workload:

- The other workloads that are contending for resources
- The resources that are in contention
- The relative impact caused by each contending workload

Impact analysis is most effective when it is used as part of a logical tuning approach to system performance. To use this approach, you set up service-level agreements that define performance target levels for critical workloads and resources. A performance monitor or an exception monitor alerts you when a given workload fails to meet the service-level targets. Once the performance problem has been identified, you need a diagnostic tool to help identify the cause of the resource bottleneck.

Workload contention

Workload contention is the condition that occurs when workloads on a system interfere with each other. When a particular workload is not performing well, the problem may not be in the workload itself, but may be caused by some other workload that is contending with it for a resource.

The impact analysis feature determines how various workloads on a system are interfering with each other. It shows which workloads are using the resources that the impacted workload needs. This analysis provides the information that can help you to reduce degradation of the monitored workloads.

The impact analysis feature analyzes workload contention from the perspective of the monitored workload. (The monitored workload is usually a favored workload with a performance problem that must be addressed without delay.) The impacting workloads can be batch jobs, started tasks, or TSO users. By focusing on resource contention from the perspective of the impacted workload, impact analysis helps you to avoid false conclusions and arrive at quick and accurate solutions for problems caused by workload contention.

Selecting a diagnostic tool

Degradation analysis (DEXAN for MVS) and impact analysis are diagnostic tools that offer complementary views of the same problem. Which one should you use? The answer depends on the kind of problem that you are experiencing and the kind of information that you are looking for.

Degradation analysis is effective as a general exception monitor, and as a diagnostic tool that deals with certain symptoms that indicate a tuning problem. Degradation analysis shows if a resource is unavailable or in contention. However, it provides no direct information about how to deal with problems resulting from workload contention. If a resource is in contention, you must gather additional information to identify the primary users of the resource.

Impact analysis is well suited for detailed analysis of workload contention problems that have been identified by DEXAN or exception analysis. Impact analysis can accurately identify the source of resource contention, because it focuses on workloads as they contend for resources.

You may find the following guidelines useful when you are deciding on the best diagnostic tool to use: If you want to gather detailed information on resource usage to help tune applications for best performance and to understand reserve capacity as indicated by idle states, use degradation analysis as your routine performance monitor. If your primary goal is to quickly solve acute performance problems, impact analysis offers you the best diagnostic capabilities.

Impact analysis displays

There are four kinds of impact analysis displays. Most impact analysis displays show you both short- and long-term impact. This perspective allows you to identify those workloads that are impacting the workload performance now, and to distinguish them from workloads that were a problem ten minutes ago, but are only marginally affecting current performance.

Workload Impact Profile

Shows a graphic profile of the impact caused by the workloads that are contending with the monitored workload.

Overview

Shows the same information in a quantitative analysis, ranking the contending workloads in order of relative impact. Each contending workload can be a batch job, a started task, or a TSO user. Although this display does not show the type of resource contention for each workload, it quickly identifies the workloads that are impacting the monitored workload.

Summary

Shows the resources that are in contention for each of the impacting workloads that are displayed in the overview.

Detail

Shows detail information for two of the possible types of resource contention: Contending I/O and Holding Enqueue. This information can help you make accurate troubleshooting decisions and take effective action to remedy acute performance problems.

Solving critical problems

Critical problems are those which seriously degrade workload performance and need to be addressed immediately. The nature of the impact source will determine which method you choose to solve the problem. Most solutions assume that the impact source is less important in the short run than the workload, and that the goal is to improve workload performance.

You can take several types of actions once you identify the impact source:

- Cancel the impact source.

If the impact source is a batch job, you can automatically restart the job later by issuing JES HOLD and QUEUE commands, followed by an OS CANCEL. (This may not be feasible for all types of jobs. Certain applications have prohibitively long shutdown times and often cannot be canceled and restarted once they have begun. For example, a database update program on a large database may not be a good candidate for this type of action.)

- Reduce service to the impact source.

You can do this by changing the impact source to a service class with less favorable service characteristics. This does not necessarily eliminate contention with the impacted workload, but is useful for those impact sources that cannot be canceled or swapped.

- Improve service to the impacted workload.

If the type of impact is in a general resource, such as CPU or storage, resetting the workload to a service class with more favorable characteristics can relieve the immediate problem. This action should be followed up with a review of the overall assignment of workloads to service classes, to avoid the problem in the future.

Types of contention

Impact analysis can identify several types of contention.

The following list describes the types of contention that can be identified by impact analysis, and suggests appropriate responses for each situation.

Contending I/O

Indicates that the monitored workload is queued for an I/O device, and that the impact source is causing the device to be unavailable. The various impact analysis displays identify the device address that is in contention. Given this information, you can move the impacted workload's data sets to a DASD volume where they can be accessed without contending with other address spaces, or reschedule the contending jobs so that they run at different times.

Another approach is to move the data set to a different location on the DASD volume in order to reduce head movement. This might be an appropriate response to a self-contention problem for data sets that were formerly adjacent and have moved apart as a result of DASD space maintenance operations.

Using CPU

Indicates that the impacted workload is waiting for CPU while the impact source is using it. This is only significant when the impact source is running at the same (or higher) dispatching priority as the impacted workload, and can be addressed by adjusting service class parameters of either the impact source or the impacted workload.

If you are already running the impacted workload at the highest available dispatching priority, the only contention for CPU should occur with system address spaces and other high-priority tasks. You may be able to improve performance of the high-priority workload through tuning, or by running the workload when system tasks are less heavily used.

For accurate results, make sure impact analysis is executing at a higher priority than the monitored workload when the wait state is Using CPU. If impact analysis executes at a lower priority, it may not get a chance to observe the monitored workload in all of its typical states, since the impact analysis address space may not be dispatched if the monitored workload is using CPU (or waiting for CPU) and no other CPU is currently available. To ensure correct relative dispatching priority, use impact analysis only in VTAM or dedicated mode.

Real storage

Real storage contention occurs when storage isolation has not been properly specified for the impacted workload and there is some degree of real storage shortage in the system, or when the storage constraint is so severe that full control of page stealing is not possible.

A frequent problem is that the paging rate of the impacted workload does not remain constant throughout the day, and certain peak periods require more storage than others. The PWSS and PPGRT parameters in the IEAIPSxx member of PARMLIB (and their CWSS and CPGRT counterparts) allow you to make some allowances for this, but they are not always effective when other address spaces are actively contending for storage. These parameters can be set to raise the target frame count so that it is protected from page stealing either in the impacted workload's private area or in the common area (PLPA or *mSA*) when the respective paging rate exceeds a threshold value.

Another way to tune for real storage contention is to lower the requirements of other workloads for real storage.

Holding enqueue

An enqueue controls access to a given resource. For example, by specifying DISP=OLD when you want to write to a data set, you are telling z/OS to issue an exclusive enqueue on that data set. As long as the resource is enqueued, other z/OS users are not allowed access to it. They must wait until you are finished with the data set and the enqueue is released.

Many applications use enqueues to obtain exclusive control over resources. If contention for enqueue resources is a persistent problem, you may need to separate the impacted workload and the impact source(s) into different environments, so that they will not compete for (and enqueue on) the resource. Using different environments may mean running certain jobs at different times, or perhaps (if the scope is LOCAL and not SYSTEMS) moving one of the workloads to a different CPU.

Mount pending

Indicates delays due to disk or tape mounts. This is considered system contention, and is not attributed to competing address spaces.

SRM delay

SRM delays are due to swaps performed to accommodate other address spaces in the same domain as the impacted workload. There are four categories of SRM delay:

- Enqueue exchange swaps (ENQ)
- Exchange swaps (EXC)
- Out and ready (MPL)
- Unilateral swaps (UNI)

To remedy the situation, you can do any of the following:

- Make the impacted workload nonswappable.
- Raise the MPL of the domain.
- Transfer other workloads out of the domain by changing them to another service class.

Holding CML lock

The monitored workload is waiting for its local lock while the impact source is holding it as a cross-memory local (CML) lock.

z/OS uses locks as a way to serialize access to resources. Each address space needs control over its local lock in order to get dispatched in the CPU. Another workload can, however, get control over that local lock by obtaining the CML lock.

When this type of contention appears, it indicates that the impacting workload has obtained the CML lock and has gained control over the workload's local lock. The monitored workload must wait until the CML lock is released before it can regain control over its local lock.

Self-contention

On group displays, self-contention is indicated by the appearance of group members shown as impact sources. When self-contention occurs, you can display contention for the individual members of the group in order to see how the members are contending with each other for resources.

These are suggestions about how to reduce contention between the impacted workload and other workloads. While there are many ways to do this, much depends on the relative priorities of the impacted workload and the impacting workloads. Once you have the kind of information that impact analysis provides, you can find appropriate ways to reduce contention at your particular installation. The following examples illustrate how you can use impact analysis in specific situations.

Performing impact analysis functions

The functions in the impact analysis feature are controlled by the IANL command.

This command enables you to perform the following tasks:

- Start and stop an impact analysis session.
- See how a favored workload is being impacted by other workloads.
- Group workloads together and see how they are being impacted as a group.
- List workloads and groups that are being monitored in the current session.
- Add or delete workloads from the session.
- Change default settings for the collection and display parameters.

IANL command syntax

IANL

Controls impact analysis functions.

Type:

Immediate

Format:

```

IANL [b]
      [Workload, [DELETE]]
      [GROUP=Groupname=(Memberlist)]
      [END/STOP/QUIT]
      [LIST[=ALL]]

```

where:

I

Is the label used to control the display format. Valid labels are:

b

Displays the overview format. (The prefix **b** indicates that column 1 is blank and you enter IANL beginning in column 2.)

S

Displays the summary format.

D

Displays the detail format.

b

Displays the impact analysis screen for the last monitored workload that was displayed.

workload

Starts monitoring and displays information for the specified workload. If no workload is specified, the default is the last workload that was displayed in the current session.

GROUP=Groupname

Selects a user-defined group of workloads for analysis. The group name is defined by the operand GROUP = group name = member list.

GROUP=Groupname=(member list)

Defines a group of workloads as an impact group, where group name is the user-defined group name, and member list is the list of members in the group. The first character of the group name must be nonnumeric.

LIST

Displays information about the last workload that was monitored.

Workload,LIST

Displays information about the specified workload.

GROUP=Groupname,LIST

Lists the members of a group. This operand displays impact information if the group is currently being monitored.

GROUP,LIST

Lists all defined groups.

LIST=ALL

Displays information about all workloads and groups currently being monitored.

DELETE

Ends monitoring for the last displayed workload.

Workload,DELETE

Ends monitoring for the specified workload.

END/STOP/QUIT

Ends the impact analysis session. All storage areas are freed except the defined groups.

If you enter IANL without a label or argument, the system displays impact information for the last workload that was displayed in the current OMEGAMON for MVS session.

Under ISPF, APF authorization is required to issue this command.

Starting and stopping an impact analysis session

Before starting an impact analysis session, you must first start the DEXAN collector. See [Chapter 12, “Degradation Exception Analyzer \(DEXAN\),”](#) on page 267 for more information on DEXAN.

To start an impact analysis session, enter the IANL command. Your first IANL command does two things:

1. It selects a workload or group for impact analysis monitoring.
2. It sets various collection and display options for the session. Some of these options can be changed during the session; others require that you set them before you begin. All of these options have default values, so you do not have to worry about them if you are just learning to use impact analysis. To override the defaults, see [“Setting defaults”](#) on page 299.

You can enter the IANL command on any part of any screen, but the top of the screen is best for optimum display results.

Once you have started monitoring a workload, you can display any of the three display types for it by adding the appropriate label, or you can display a Workload Impact Profile by adding the IPRO command. See [“The Workload Impact Profile”](#) on page 298 for more information on the IPRO command. If you enter the IANL command without a workload name, the system remembers the last workload that you selected during the session and generates a display for that workload.

To stop an impact analysis session, add the END keyword to the IANL command. Stopping an impact analysis session stops impact analysis monitoring and allows you to reset certain collection options. It does not terminate OMEGAMON for MVS; you can enter and re-enter impact analysis as many times as you like during a single OMEGAMON session.

Selecting a workload

To begin monitoring a single workload, enter the IANL command followed by the name of the workload or address space to be monitored.

For example, to begin monitoring a workload called PAY0022, enter the following command:

```
IANL PAY0022
```

When you first enter this command, the system begins monitoring but does not show you a display. Press Enter to generate a realtime impact analysis display. Since the command is not commented out after execution, press Enter (or let the screen refresh in automatic update mode) to update the display.

Each time you select a workload, it is monitored until the session is terminated, the job ends, or you stop monitoring. To stop monitoring a workload, add the DELETE keyword to the command, separating the keyword from the job name by a comma. For example, to stop monitoring PAY0022, enter the following command:

```
IANL PAY0022,DELETE
```

Defining a group workload

You can define a group of workloads to be monitored as a single workload. To define the group workload, use the GROUP keyword with the IANL command.

For example, to combine the TSO users TSO1, TSO2, and TSO3 into a reporting group called TSO33, enter the following command:

```
IANL GROUP=TSO33=(TSO1,TSO2,TSO3)
```

Follow the GROUP keyword with the name of the reporting group, and then with the job or started task names in the group. The list of members must be contained within parentheses, and must be separated by commas or blanks.

The only limit on the number of workloads that can be defined as a group is the number of job names you can fit on one input line. You can avoid running out of space in the line by using masks (that is, task names that include wildcard characters) to define groups. For example, if you specify:

```
IANL GROUP=TS033=(TS0*)
```

all TSO users that begin with TSO are selected for the group.

When you define a group, the command is automatically commented out with the greater-than sign. In this example, the display looks like this:

```
> IANL GROUP=TS033=(TS01,TS02,TS03)
+      << IA0209 Group Has Been Defined>>
```

The group definition lasts until the end of the current OMEGAMON session. You can avoid the need to redefine group workloads by creating a screen space to retain group workload definitions. See [Chapter 4, “Screen spaces,”](#) on page 71 for more information on screen spaces.

To monitor this user-defined group, issue the group selection command, as shown in the next section.

Selecting a user-defined group

To select a user-defined group, use the GROUP keyword with the IANL command.

For example, to monitor the group TSO33 (as defined in the previous section), enter the following command:

```
IANL GROUP=TS033
```

The first time you issue this command, monitoring begins for the specified group. Since the command is not commented out after execution, you only need to press Enter (or let the screen refresh in automatic update mode) to update the display.

Once you define a group, monitoring continues automatically until the session is terminated or you stop monitoring. You stop monitoring the group by adding the DELETE keyword to the command. For example, to stop monitoring TSO33, enter the following command:

```
IANL GROUP=TS033,DELETE
```

After you delete a group, you can restart monitoring for that group without redefining its members. For example, to restart monitoring group TSO33, enter the following command:

```
IANL GROUP=TS033
```

OMEGAMON remembers group definitions for the duration of the OMEGAMON session, even if you terminate impact analysis monitoring with the IANL END command.

Listing workloads and groups

You can list jobs and groups that are currently being monitored with the LIST keyword.

To display the last job or group selected for monitoring, use LIST without an operand.

```
IANL LIST
```

For example, if the last workload you have selected for monitoring is PAY0022, the display looks like this:

```
IANL LIST
+ PAY0022 >>Collection started 12:21:12, elap 25:41 MN, short=201, long=1537 <<
```

This display shows the time that monitoring began for the workload, how long monitoring has been going on, and the number of observations in the short-term and long-term intervals.

To display a list of all jobs and groups currently being monitored, add the ALL operand.

```
IANL LIST=ALL
```

The equal sign between LIST and ALL is required.

The list of all workloads being monitored looks similar to the following:

```
IANL LIST=ALL
+ PAY0022 >>Collection started 12:21:12, elap 25:41 MN, short=201, long=1537 <<
+ BATXX >>Collection started 12:21:35, elap 25:18 MN, short=185, long=1430 <<
+ TS021 >>Collection started 12:36:25, elap 10:28 MN, short=335, long=715 <<
+ TS033 >>Collection started 12:40:06, elap 6:47 MN, short=58, long=452 <<
+ MEMBERS:
+ TS01
+ TS02
+ TS03
```

Since TS033 is a user-defined group, the display shows the members of the group.

The impact analysis display format

The impact analysis display has four parts.

```
1 [IANL PAY0022 IAST=2.2 IATH=5 IACS=1 IA CL=5
2 [Productive/Contention/Other | 10.8% 83.7% 5.5% | 38.7% 52.5% 8.8%
3 [Contending Address Spaces |Short (3:24M,88) |Long (13:24M,361)
@-----+-----+-----+-----+-----+-----+-----+-----+
| *Total Contention* *83.7*|----->>>>|*52.2*|----->>>>
| *Self-Contention* 10.2|-> . . . . . | 7.9|-> . . . . .
4 | WORK0033 (J1208) 37.5|-----> . . . . . | 0.0| . . . . .
| TESTCICS (S1366) 13.3|--> . . . . . | 11.0|-> . . . . .
| USER01 (T0934) 0.0| . . . . . | 15.3|--> . . . . .
% System 22.7|-----> . . . . . | 18.3|-----> . . . . .
```

Figure 69. Impact analysis display format

1. Input command and options

In the example shown in Figure 69 on page 294, the input command is IANL PAY0022, which requests an overview display for workload PAY0022. The remainder of the line contains information about the collection and display parameters in effect. For this command, the parameters are set as follows:

- The sampling interval (IAST) is 2.2 seconds.
- The plot threshold (IATH) is 5%.
- The short term interval (IACS) is 1 minute.
- The long term interval (IA CL) is 5 short-term intervals (about 5 minutes).

These parameters and their default settings are discussed in “Setting defaults” on page 299.

The next two lines, the productivity analysis line and the interval description line, should be read as a pair. When read together, these two lines give you a general idea of how reliable your data is, and how the workload (or group) is being impacted by other address spaces in the z/OS system. Long-term and short-term contention are shown side-by-side; long-term figures are on the right side of the display under the Long heading, and short-term figures are in the center of the display under the Short heading.

2. The productivity analysis line

Impact analysis analyzes workload productivity by sampling the execution state of the monitored workload at periodic intervals. By taking a large number of samples, it is possible to draw valid inferences about how a workload is spending its time.

The productivity analysis line of the impact analysis display provides a breakdown of how the impacted workload has been spending its time, in both the long and short term.

Three categories of time are represented on impact analysis displays:

Productive

Time spent doing desirable activities such as using CPU or active I/O. In this example, the workload has been spending 10.8% of its time in the short term doing productive work, and 38.7% in the long term.

Contention

Time spent waiting for other resources that are being used by other z/OS address spaces. This type of wait is referred to as a chargeable wait, and is associated with the following execution states:

- CPU waits (waiting for dispatch by z/OS) contending I/O
- Queued I/O
- Reserves (charged only to system)
- Tape or disk mount pending
- Private area (local) page waits
- Common area page waits
- Waits for swap page-ins
- Enqueue waits
- SRM delays
- Swaps due to real storage shortages
- Cross-memory local lock waits (CML)

In this example, the workload has been spending 83.5% of its time in the short term waiting for contending address spaces, and 52.5% in the long term.

Other

Time during which the workload is waiting, but the wait time is not attributable to another address space. For example, a tight CPU loop unnecessarily executes instructions, wasting time that would be put to better use processing data. Although the address space stuck in a tight loop is degraded, no other address space is responsible.

This type of wait is referred to as a nonchargeable wait, and is associated with the following execution states:

- Waiting for MSS staging
- Local or global lock waits (except CML)
- SRM delays (RTO)
- Swaps due to terminal output waits
- Swaps due to detected waits
- Requested swaps
- Transition swaps

In the example, this type of wait time accounts for 5.5% of the short term and 8.8% of the long term.

Another type of time that is associated with productivity, but is not reported on impact analysis displays, is referred to as idle time. Idle time is time in which the monitored workload is not doing

work, but is not being prevented from doing so. Idle time is associated with the following execution states:

- ECB waits
- ECB waits with STIMER
- STIMER waits
- Long wait swaps
- HSM waits
- JES2 waits

By looking at the productivity analysis in the long and short term, you can understand how workload productivity is changing. In this example, contention has been more severe in the short term than in the long term, accounting for 83.7% of the non-idle time, as opposed to 52.5% in the long term. Similarly, productive time has dropped from 38.7% in the long term to 10.8% in the short term.

3. The interval description line

The interval description line, which immediately follows the productivity analysis line, shows the length of the short and long term intervals, and the number of observations in each interval.

The short term interval side of the display shows the length (in minutes) of the short term interval, and the number next to it indicates the number of observations made during that period.

The long term interval side of the display shows the length (in minutes) of the long term interval, and the number next to it indicates the number of observations made during that period.

In this example, the short term interval contains information collected over the past 3 minutes and 24 seconds, during which there were 88 observations. Similarly, the long term interval spans 13 minutes and 24 seconds, during which 361 observations were made.

Since the reliability of the data improves with the number of observations, it is helpful to note the number of observations in the intervals. In general, you should wait until the display shows at least 30 observations per interval before relying on this data.

4. Display body

The body of the display shows various contenders and the degree to which they are impacting the workload. The number next to each impact source indicates the percent of the total non-idle time that the workload was found waiting for one or more resources while that contender was using it.

The arrows to the right of the numbers provide graphic representations of the data in the following ways:

- If the contention is less than thirty percent, the arrow is made up of hyphens only.
- If contention is greater than thirty percent, the arrow between thirty and sixty percent is made up of equal signs.
- Contention greater than sixty percent is designated by greater-than (>) signs.

On extended color terminals, the arrows are displayed in color. The colors for the three contention levels are green, yellow, and red, respectively. For information about turning on extended color, see [Chapter 5, "The User Profile Facility," on page 77](#).

The first line of this section shows total contention. (Total contention is also shown as part of the productivity analysis line. It is provided here for easy comparison with other lines in the display body.) The total contention shown on this line may not be equal to the sum of the contention from all impact sources shown underneath it. This is because only the main impact sources are shown on the default display (only those that account for 5% or more of the non-idle time are included). You can adjust this threshold with the IATH command, as described in ["Setting defaults" on page 299](#).

The next three lines of the sample display show address spaces that are competing with PAY0022 for resources. The display lists competing address spaces in order of decreasing short term impact. The most important contenders are shown first.

Below the short-term contenders, the display lists long-term contenders that are not causing any short term contention. Long-term contenders are also listed in order of decreasing impact.

The example shows three impact sources: a batch job (WORK0033), a started task (TESTCICS), and a TSO user (USER01). Next to each name, the display shows the JES2 job number of the impact source. The job numbers each begin with a letter that denotes the type of job:

- J**
identifies a batch job
- S**
indicates a started task
- T**
identifies a TSO user

In this example, WORK0033 is the largest short term impact source, accounting for 37.5 percent of non-idle time. However, the long term display shows USER01 to be the major impact source over a longer period of time, even though it is not currently impacting the monitored workload.

The last line of the display shows system contention. This is primarily due to contention that is detected but cannot be attributed to any particular address space. You can minimize the system contention value by running OMEGAMON at a high dispatching priority. The following section provides more information on system contention.

How resource contention is shown

A contending workload can impact the performance of a monitored workload in several ways. Impact analysis shows resource contention (in the summary and detail displays) in one or more categories.

In addition to contending workloads, impact may also be attributed to System, which means that impact analysis is unable to identify a specific address space as the source of contention. This may be because the contention stopped before impact analysis was able to analyze it, or because the nature of the contention is such that no single address space is responsible. System contention always appears as the last item in an impact analysis display, regardless of its severity.

Impact analysis attributes relative impact to System in different ways, depending on the type of contention that is being analyzed. Impact analysis determines the amount of impact caused by each contending workload as follows:

1. Impact analysis examines each execution state of the monitored workload and determines whether the state is active, idle, or contended.
2. If the execution state is contended, impact analysis identifies the offenders that caused the contention.
3. Each contention state is assigned a point value, which is distributed among the impacting workloads.
4. If a contention state cannot be assigned to an impacting workload, it is assigned to System.

The following list shows how impact analysis assigns relative impact to System for each category of contention.

CPU contention

The impact caused by CPU Contention (the monitored workload is waiting for CPU) is assigned to System if impact analysis cannot find a number of impacting workloads equal to the number of online CPUs. If the monitored workload is found active on the CPU, System is considered the offender, because it is assumed that the contention cleared before it could be analyzed.

I/O contention

If the monitored workload is queued for I/O and no address space is currently active on the UCB, the impact is charged to System. If the monitored workload is active on the device, the impact is charged as self-contention, since it is possible for an address space to be active and queued for I/O on the same UCB. For MVS/370 systems, contention caused by Reserved I/O is charged to System.

SRM delays

When a monitored workload is found swapped out by the SRM either unilaterally or in an exchange, or is swapped out and has ready work, impact analysis identifies the address spaces in the monitored workload's domain. Relative impact is assigned to System if the monitored workload is not swapped out any more when the routine is invoked, if the monitored workload's domain has a target MPL of 0, or if the analysis routines do not identify an address space.

Storage delays

When a monitored workload is found swapped out by the SRM because of a shortage of real or auxiliary storage, impact analysis computes the total number of storage frames online and identifies address spaces using more than 10% of storage. Relative impact is distributed among address spaces using more than 10% of storage based on the percentage of storage they use. Leftover impact is attributed to System.

Paging Delays

When a monitored workload is found waiting for a page (common or private), impact analysis determines if there is a shortage of real storage in the system based on the available frame queue and threshold. If there is a real storage shortage, that analysis routine is invoked and relative impact is assigned as in storage delay counts. If there is no storage shortage, the page delay time is examined, and if higher than 25 ms., the I/O contention routine is called to identify contention for the paging I/O device. If paging delay is not the problem, the real-storage shortage analysis is used.

Enqueue delays

When a monitored workload is found waiting for an enqueue, impact analysis identifies the address spaces holding the enqueue. If the monitored workload requested shared access, the address spaces on the list of waiters and requesters above the monitored workload holding or waiting for exclusive use are counted as impacting workloads. If the monitored workload requested exclusive control, all address spaces on the list above the monitored workload are counted as impacting workloads. If the monitored workload has the enqueue when the analysis is performed, if enqueue was requested from another system, or if no impacting workloads are found, the contention is charged to System.

Lock delay

When a monitored workload is found waiting for the CML lock, the holder of the lock is identified. Impact is attributed to System if the lock holder's address space is gone from the system when the count is recorded.

Mount pending

Mount Pending is always attributed to System.

The Workload Impact Profile

The Workload Impact Profile is another form of overview display in impact analysis that shows a graphic illustration of current contention on your system.

IPRO

Generates the Workload Impact Profile.

Type:

Immediate

Format:

```
IPRO [IANL  
      [Workload]  
      [GROUP=Groupname]]
```

IPRO does not support the labels S and D, or the suffix PD.

Note the following guidelines for using this command:

- IPRO cannot be used to initiate monitoring. You must first start monitoring the job or group with the IANL command.
- If you do not specify a jobname or group name, IPRO defaults to the last job or group specified with IANL.

- If you do not specify IANL, the operand defaults to the last one issued with IPRO.

The IPRO command remembers its last set of operands, and if it is entered alone, it uses the last operands to construct the profile.

Contending workloads are shown in order of decreasing severity, including self-contention and system contention.

The profile displays only the long-term wait condition figures available with the IANL command, and not the short-term ones, to ensure a statistically significant display.

For terminals with extended color in effect, the Workload Impact Profile appears in the colors set for levels 5, 6, and 7. The graph itself appears in the level 5 color, the impact source labels are in the level 7 color, and the lines pointing from the impact source labels to the graph are in the level 6 color. For information about turning on extended color and setting the color for each level, see [Chapter 5, “The User Profile Facility,”](#) on page 77.

Before you use the Workload Impact Profile, be sure that the impact analysis collectors have been running for several minutes in order to establish a statistically representative sample. For information about how to start the collectors, see [“Starting and stopping an impact analysis session”](#) on page 292.

The Workload Impact Profile is most useful when the favored workload is experiencing response time problems. When response time is acceptable, the profile provides a picture of contention that is noncritical, and no tuning is required. If response time worsens due to contention from other workloads, the Workload Impact Profile may also change, indicating a possible cause of the response time problem.

The impact analysis time-out facility

While the impact analysis collector is active it is consuming resources. Since it is possible to start impact analysis collection and then move on to other kinds of analysis without turning the collector off, the impact analysis time-out facility automatically terminates the collector after a period of time. This time period is equal to the short-term collection interval multiplied by the number of short-term intervals in the long-term interval. For example, if the short-term interval is one minute and the long-term interval is five short-term intervals (the defaults), impact analysis times out after five minutes.

You can modify the time-out period with the IATO command.

IATO

Specifies how long the impact analysis collector remains active.

Type:

Immediate

Format:

IATOnn

The variable *nn* is the number of minutes the impact analysis collector remains active after an impact analysis command is issued. The valid range is 0 (meaning that impact analysis will not time out), through 99 minutes.

When the period of time you specify with the IATO command elapses without a subsequent impact analysis command, the collector terminates, and frees all data areas except the group definitions and workload memory block. Any impact analysis command, including environmental commands such as IAST and IACS, begins a new time-out countdown.

Setting defaults

Impact analysis comes with a number of defaults to help you use the collection and display features quickly and easily. As you gain experience with the product, you may want to adjust some of these defaults to meet the specific conditions of your installation.

You can make the following adjustments to customize the product:

- Determine how often the status (execution state) of a monitored workload is checked.
- Determine how often the short-term and long-term interval collection buckets are cleared or reset.
- Set a maximum number of workloads that can be monitored with the IANC command.
- Set parameters for enqueue collection (which are the most costly in terms of CPU utilization) independently of other types of contention monitoring.
- Adjust the number of contenders that are shown on the impact analysis displays based upon the severity of the impact they are causing.

Use the commands that are summarized in [Table 18 on page 300](#) to change default settings.

Command	Description	Operand	Default
IAST[nn]	Sets the sampling interval in tenths of a second. If entered without an operand, the current value of the interval is displayed. The default is .9 seconds. This command can be used while the collector is running.	1–99	.9 seconds
IACS[nn]	Sets the interval at which the short-term buckets are cleared (in minutes). If entered without an operand, the current value of the interval is displayed. The default is 1 minute. Each execution of the command (with or without an operand) automatically clears the short-term buckets. This command can be used while the collector is running.	0–99	1 minute
IACL[nn]	Sets the interval at which the long-term buckets are reset (in number of short-term intervals). If entered without an operand, the current value of the interval is displayed. The default is 5 short term intervals. Each execution of the command (with or without an operand) automatically resets the long term buckets. This command can be used while the collector is running.	0–99	5 short- term intervals
IANC[nn]	Sets the maximum number of workloads that can be monitored. If no operand is entered, the current value is displayed. The default is 5. Although the collector must be stopped to change this parameter, the current value can be displayed while collection is ongoing.	1–5	5
IANQ[nn] [ON] [OF]	Sets the enqueue sampling interval as a multiple of the normal sampling interval. The ON and OF operands enable and disable enqueue data collection. By default, enqueue sampling is disabled. If enqueue sampling is enabled, the default multiple is 10. Although the collector must be stopped to change this parameter, the current value can be displayed while collection is ongoing.	1–10 ON OF	OF
IATH[nn]	Determines the number of contenders shown on the IANL display, based upon the severity of the impact. By default, only contenders that account for 5% or more of the total non-idle time are shown. The operand to this keyword sets the display threshold as a minimum contention amount (expressed as a percentage of total contention). For example, IATH03 sets the threshold at 3%. If no operand is entered, the current value is displayed. If the operand is set to 0, all contenders are shown. This command can be used while the collector is running.	0–99	5%

Table 18. Commands to set impact analysis defaults (continued)

Command	Description	Operand	Default
IATO[nn]	Specifies how long the impact analysis collector remains active after you issue an impact analysis command. By default, the time-out facility automatically terminates the impact analysis collector after a time period equal to the number of minutes in a short-term interval multiplied by the number of short-term intervals in the long term interval. For example, if the short term interval is 1 minute and the long-term interval is 5 short-term intervals (the defaults), Impact Analysis times out after 5 minutes. The valid range is 1 to 99 minutes. The value 0 prevents the time-out.	0–99	one long-term interval

Chapter 14. Applications tuning

The applications tuning facility monitors applications running in an address space and shows CPU resource consumption.

CPU resource consumptions is shown by the follow categories:

- Task
- Load module within a task
- CSECT within a load module or
- Offset within a CSECT

Applications tuning facility INSP major command

The INSP major command displays the current sampling parameters and status of the applications tuning facility.

Type:

Major

Format:

INSP JOB(cccccccc) SAMPLES(nnnn) | NOLIMIT INTERVAL(nnn) STOP | START

When you enter the INSP command beginning in column 2, the cursor automatically moves to column 7 to accept an operand. Use one of the following operands:

JOB(cccccccc)

The variable *cccccccc* specifies the address space INSP monitors. This can be a job name, * (the current address space), or the ASID value in decimal.

SAMPLES(nnnn)

Number of samples to gather. The default is 3000. If you set the sample number to 0, it has the same effect as the NOLIMIT operand.

NOLIMIT

Specifies sampling to continue until you enter the STOP command.

INTERVAL(nnn)

The variable *nnn* is the duration of the interval (in milliseconds) between samples. The default is 10 milliseconds.

STOP

Stops sampling immediately. You can use this parameter to terminate sampling before INSP reaches the specified sampling limit.

START

Begins sampling and displays results. Using the action character (-) in column 1 is the same as using this operand.

The INSP command requires APF-authorization.

To use the applications tuning facility, complete the following procedure:

1. Issue the INSP command.

```
INSP
```

The following display appears:

```

INSP
+ Current sampling options:
+ JOB(USER1) ASID is 38 Sampling is not active
+ SAMPLES(200) INTERVAL(100)

```

- To begin sampling, enter INSP START or the action character (-) in column 1.

```
INSP START
```

Sampling begins and its progress appears:

```

INSP
+ Current sampling options:
+ JOB(USER1) ASID is 38 Sampling is active
+ SAMPLES(200) INTERVAL(100)
+ Samples taken = 25; samples used = 15

```

The figure given for samples taken includes both valid and invalid samples. Samples used reflects valid samples. An invalid sample may result from the address space being swapped out or not accumulating CPU.

- INSP completes the sampling when the SAMPLES limit is reached or when you enter the STOP parameter:

```
INSP STOP
```

The following display appears:

```

INSP
+ Current sampling options:
+ JOB(USER1) ASID is 38 Sampling is not active
+ SAMPLES(200) INTERVAL(100) 200 samples have been taken
+ Samples taken = 25; Samples used = 15

```

Applications tuning minor commands

The ICPU minor command displays the CPU activity within an address space. The MLIB specifies the INSP module libraries.

ICPU minor command

The ICPU minor command displays the CPU activity within an address space.

CPU activity information is shown in real time during the progress of a job.

Type:

Minor of INSP

Format:

ICPU nn TCB(ccc) LMOD(ccc) CSECT(cccc) OFF(cccccc) GR(ccccc) aaaaa

nn

Skips first nn lines of display. This can be a number from 1 to 99999. The default is 0.

TCB(ccc)

Specifies which task (ccc) to monitor. The default is *, which specifies all tasks.

LMOD(ccc)

Specifies load module (ccc) to monitor. The default is *, which specifies all load modules.

CSECT(cccc)

Specifies CSECT (cccc) to monitor. The default is *, which specifies all CSECTs.

OFF(cccccc)

Specifies the first offset (cccccc) of a CSECT to monitor. The default is X'0'.

GR(ccccc)

Specifies the granularity (ccccc) of the CSECT display. For example, GR(10) offsets appear X' 10' bytes apart.

When you monitor at the CSECT level, OMEGAMON calculates the default so that the display just fits the physical screen.

aaaaa

Specifies the order of the CSECT display.

HOT

Descending order of CPU consumption

NOHOT

Instruction addresses (default)

The following shows how to proceed through the different levels of the ICPU display.

When you first enter ICPU with no operands, TCB(*) appears beside it to indicate that all tasks are being monitored. The first line of the ICPU display shows % CPU utilization. Succeeding lines show tasks in order of their addresses (the default). The following figure shows the MONITOR task using 100% of the CPU.

```

INSP
+   Current sampling options:
+   JOB(PROBJOB) ASID is 88      Sampling is active
+   SAMPLES(3000)   INTERVAL(10)
+   Samples taken = 3000;  samples used = 7
ICPU TCB(*)
+<.: % CPU Utilization      +-----20-----40-----60-----80-----100
+ 1 61DA48 IEAVAR00         |
+ 2 61D258 IEESB605         |
+ 3 60A158 IEFIIC          |
+ 4 56FC48 MONITOR 100.0 |=====>
+ 5 61D510 IEAVTST         |
+ 6 572E28*MONITOR2        |

```

Note: On a terminal with extended mode graphics, reverse video blanks appear as the plot symbol.

The asterisk (*) between address 572E28 and module MONITOR2 indicates a task that was present during earlier samples, but is no longer present.

To further investigate the MONITOR task, use one of the following methods:

- Change the TCB(*) to TCB(4) OR move the cursor to the 4 field and press Enter.
- Change the TCB(*) to TCB(56FC48) OR move the cursor to the 56FC48 field and press Enter.
- Change the TCB(*) to TCB(MONITOR) OR move the cursor to the MONITOR field and press Enter.

ICPU then displays the resource consumption of the load modules associated with the task:

```

INSP
+   Current sampling options:
+   JOB(PROBJOB) ASID is 88      Sampling is active
+   SAMPLES(3000)   INTERVAL(10)
+   Samples taken = 3000;  samples used = 7
ICPU TCB(56FC48) LMOD(*)
+<.: % CPU Utilization      +-----10-----20-----30-----40-----50
+ 1 125238 MONITOR 19.30 |=====>
+ 2 254320 MONDSPL0  .00 |
+ 3 3557A8 MONTEMP  .00 |
+ 4 60A158 MONGRPH0 34.60 |=====>
+ 5 65CF48 MONDATA0 46.10 |=====>
+ 6 C70FD3 ISGPREP +15.40 |=====>

```

The plus sign next to module ISGPREP means that CPU time was accumulated under another address space ID.

Note that OMEGAMON adjusts the CPU utilization scale to maximize the use of the plots, following the pattern 100%, 50%, 10%, 5%, 1%, and so on.

You can then examine the CPU consumption of the MONITOR load module, broken down into CSECTs, by one of the following methods:

- Change the LMOD(*) to LMOD(1) OR position the cursor in the 1 field and press Enter.
- Change the LMOD(*) to LMOD(125238) OR position the cursor in the 125238 field and press Enter.
- Change the LMOD(*) to LMOD(MONITOR) OR position the cursor in the MONITOR field and press Enter.

ICPU then displays the resource consumption of the CSECTs associated with the load module. The dots underneath CSECT 4 indicate that data is omitted from the figure.

```

INSP
+ Current sampling options:
+   JOB(PROBJOB) ASID is 88   Sampling is active
+   SAMPLES(3000)   INTERVAL(50)
+   Samples taken = 3000; samples used = 7
+ ICPU TCB(56FC48) LMOD(MONITOR) CSECT(*)
+<.: % CPU Utilization  +----+----.1----+----.2----+----.3----+----.4----+----.5
+ 1 125238 MAIN      0.412 |=====>
+ 2 134020 WASTE     0.184 |=====>
+ 3 1362000 LIE      0.349 |=====>
+ 4 142450 MISTAKE   0.440 |=====>
.
.
.

```

You can then examine the CPU usage of the MAIN CSECT by one of the following methods:

- Change the CSECT(*) to CSECT(1) OR position the cursor in the 1 field and press Enter.
- Change the CSECT(*) to CSECT(125238) OR position the cursor in the 125238 field and press Enter.
- Change the CSECT(*) to CSECT(MAIN) OR position the cursor in the MAIN field and press Enter.

ICPU then displays the resource consumption of the CSECT broken down into addresses. At this level, ICPU displays the default values for OFF (which is 0) and GR (which is 20); ICPU adjusts the value for GR so that the display exactly fits your physical terminal screen. To look at other parts of the CSECT, enter different values for OFF.

```

INSP
+ Current sampling options:
+   JOB(PROBJOB) ASID is 88   Sampling is active
+   SAMPLES(3000)   INTERVAL(50)
+   Samples taken = 3000; samples used = 7
+ ICPU TCB(56FC48) LMOD(MONITOR) CSECT(MAIN) OFF(0) GR(20)
+<.: % CPU Utilization  +----+----.1----+----.2----+----.3----+----.4----+----.5
+ 1 125238 +000000      |
+ 2 125248 +000020  0.018 |=>
+ 3 125268 +000040  0.354 |=====>
+ 4 125288 +000060  0.021 |=>
.
.
.

```

The dots below address 4 indicate that data is omitted from the figure.

The less-than symbol (<) in column 2 of the plot line lets you navigate through ICPU displays in reverse. Simply move the cursor to this position and press Enter to move, for example, from the CSECT level up to the load module level display.

If the load module name displays as \$PRIVATE, then use the MLIB command to specify additional libraries to be used to resolve the CSECT maps for the programs being reported on. If ICPU is unable to determine the load module name, *-UNKN-* is displayed.

MLIB minor command

The MLIB minor command specifies the INSP module libraries.

Type:

Minor of INSP

You can use MLIB to update the list of libraries used to resolve CSECT maps for programs. Several conditions must be met for OMEGAMON to resolve CSECTs:

- One or more libraries must be specified in the library list.
- One of the libraries must contain a load module that matches the name of the selected load module.
- The directory characteristics must match the CDE/LPDE of the in-storage module.
- The library load module must not be marked not-editable.

If any one of these conditions is not met, OMEGAMON treats the whole load module as one CSECT named \$PRIVATE.

Format:

```
MLIB ADD(ccccccc, ...) DEL(ccccccc, ...|nn, ...)
```

ADD(ccccccc,...)

Adds one or more libraries (ccccccc) to the list by data set name.

DEL(ccccccc,...|nn,...)

Deletes one or more libraries from the list by name (ccccccc) or number (nn). Data set numbers appear when you enter MLIB without any parameters.

Examples

If you enter the MLIB command without any arguments, the current list of libraries appears:

```
INSP
+   Current sampling options:
+   JOB(PROBJOB) ASID is 88      Sampling is not active
+   SAMPLES (3000)  INTERVAL(10)
MLIB
+   SEQ  DSNAME          VOLUME
+   1    SYS1.LINKLIB    MP217X
+   2    SYS1.LPALIB    MP217X
```

To add a library to the list, use the ADD keyword:

```
MLIB ADD(SYS1.USERLIB)
```

The result is:

```
INSP
+   Current sampling options:
+   JOB(PROBJOB) ASID is 88      Sampling is not active
+   SAMPLES (3000)  INTERVAL(10)
MLIB
+   SEQ  DSNAME          VOLUME
+   1    SYS1.LINKLIB    MP217X
+   2    SYS1.LPALIB    MP217X
+   3    SYS1.USERLIB    MP217X
```

To delete a library from the list, use the DEL keyword:

```
MLIB DEL(SYS1.LPALIB)
```

The result is:

```
INSP
+   Current sampling options:
+   JOB(PROBJOB) ASID is 88      Sampling is not active
+   SAMPLES (3000)  INTERVAL(10)
MLIB
+   SEQ  DSNAME          VOLUME
+       1  SYS1.LINKLIB  MP217X
+       2  SYS1.USERLIB  MP217X
```

Appendix A. Automating and logging features

OMEGAMON can automatically invoke displays, initiate action, and log information in response to exception conditions (that is, if an exception message is triggered), or at specific times during the day. When the sequence is finished, OMEGAMON then resumes normal operation, all without manual intervention.

There are three event- and time-driven features:

XLF (exception logging facility)

This feature automatically time-stamps and logs exception messages for your review. It enables you to correct intermittent performance problems by documenting the frequency and severity of system-wide exceptions.

ASF (Automatic Screen Facility)

This feature automatically invokes a predefined screen space when a given 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 perform other options.

TSF (Timed Screen Facility)

This feature automatically invokes screen spaces at times or time intervals you specify. Many sites use TSF to spin off copies of the REPORT or of XLFLOG files to the printer. In general, you can use the TSF facility to automate many day-to-day housekeeping routines.

XLF, ASF, and TSF all require DEXAN running in dedicated or VTAM automatic update mode. They do not function with DEXAN running under TSO. Also, XLF captures WSHI address space exceptions only.

Controlling automating features

It is important for you to be familiar with how exception analysis works and how to create screen spaces before you try to use the event- and time-driven features.

If you are operating OMEGAMON in menu mode, choose the PROFILE option of the Main Menu. From there, 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, as well as define entries for TSF.
- **LOGGING** to turn the log on and off, and to spin the log out to the printer.

XLF commands

The XACB, OPTN, and /XLF OUT commands are used to set up and operate the exception logging facility (XLF).

The commands that you need to set up and operate XLF are:

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 310.

OPTN

Turns the XLF feature ON and OFF (with the XLF keyword).

/XLF OUT

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 /XLF OUT INFO-line command.

ASF commands

The XACB, OPTN, and the /LOG OUT commands are used to set up and operate the Automatic Screen Facility (ASF).

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 310.

OPTN

Turns the ASF feature ON and OFF (with the ASF keyword).

/LOG OUT

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 /LOG OUT INFO-line command.

TSF commands

The .TSF, OPTN, and /LOG OUT commands are used to set up and operate the Timed Screen Facility (TSF).

The commands that you need to set up and operate TSF are:

.TSF

Defines entries for TSF. This command is described in [“Setting parameters for TSF operation”](#) on page 311.

OPTN

Turns the TSF feature ON and OFF (with the TSF keyword).

/LOG OUT

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 /LOG OUT INFO-line command.

Setting parameters for XLF and ASF operation

The parameters for XLF and ASF operation are set with the XACB command. XACB allows you to activate the XLF and ASF features for all occurrences of any given exception, or only if the exception persists for a specified number of cycles. You can also set a limit on the number of times a given exception invokes 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 displayed with the XACB command.

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

The XLF and ASF parameters are:

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 (*nn*) XLF and ASF are invoked if the exception occurs. If you specify `Limit=00`, no events are logged. If you specify `Limit=NONE`, XLF and ASF are invoked each time the exception occurs. The parenthetical number to the right of this parameter indicates the remaining number of times that the exception will be logged. You can reset `Limit` 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 would be logged only once as specified with the `Persist` parameter.

Persist

Logs the exception message and invokes the ASF screen spaces when the condition persists for *nn* consecutive OMEGAMON cycles. After it has reached the threshold, the message is logged only once (or ASF is invoked only once) 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 invoke if ASF is in effect (`Auto=ON`), and specifies that the output of the ASF screen spaces is to be logged. The OMEGAMON REPORT log automatically turns on when the exception trips and screen space logging starts.

SS

Specifies the screen space to invoke if ASF is in effect. It does not turn on the log.

The following is an example of defining the XCSA exception for ASF with logging:

```
XACB
:XCSA
+ DISPLAY Parameters:      THRESHOLD Parameters:      XLF Parameters:
:   State=ON              Threshold=85                 Auto=ON
:   Group=OP              Display=Red                  Log=OFF
:   Bell=ON               Attribute=NONE              Limit=20
+ BOX Parameters:         CYCLE Parameters:           Repeat=NO
:   Boxchar='+'          ExNcyc=0                    Persist=5
:   Boxclr=CLR2          Stop=0 (7)                  SL=STORAGE
:   Boxattr=BLINK        Cumulative=5
```

Setting parameters for TSF operation

The Timed Screen Facility (TSF) schedules certain screen spaces not on an exception basis, but rather at specified times of day or at specified intervals. For example, you may want to turn on bottleneck analysis automatically at 2:00 PM every weekday, and then turn it off again at 2:15. You may also want to invoke a screen space at regular intervals, such as every hour.

To use TSF, follow this procedure.

1. Create any screen spaces you want TSF to invoke.

You can use the `.SGO` or `.FGO` command to chain screens together, and `.RTN` to end the cycle. For information about creating screen spaces in TSF, see [“Using screen spaces in ASF and TSF” on page 313](#).

2. Turn on TSF with the `TSF` keyword of the `OPTN` command.
3. Use the `.TSF00` command to enter the names of the screen spaces and the times or time intervals when you want them invoked.

The `.TSF00` command displays the current status of the TSF feature (ON/OFF), and lists all entries. The TSF table is shipped with 99 blank entries. For example:

```
.TSF00  0N
      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
```

Enter .TSFnn to display entry *nn* in the TSF table. To define an entry, type .TSF followed by the number of the entry. For example, type .TSF01 to produce:

```
.TSF01  Time=0000  SS=*NONE*  DAY=DAILY
```

You can type the new entry over the current entry, press ENTER, and the value is reset.

If you want to change an entry that doesn't appear on the physical screen, you can specify an argument to skip *nn* entries. Type

```
.TSF00  20
```

to display entries 21 through 99 in the TSF table, skipping the first 20.

Following are the keywords and valid entries.

Time

Specifies the time of day (from 0000 to 2400) to invoke the screen. TIME=*+nn* invokes the screen every *nn* minutes.

Note: The screen space will not execute while the TIME=*+nn* entry remains on your current screen.

SL or SS

Specifies the screen space to invoke if TSF is in effect. SS specifies the screen space to invoke, 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 through 31). The default value is DAILY. You may abbreviate the input as long as it is unique, and as long as the day of the week is recognized.

You may specify day combinations by enclosing the names of the days within parentheses, and by separating each day with either a comma or a blank.

This 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
```

TSF01

Executes and logs screen space WENDSHFT at 6:00 PM daily.

TSF02

Executes screen space WEEKSTRT at 6:00 AM every MONDAY.

TSF03

Executes and logs screen space NOONTIME at 12:00 PM every Thursday and Friday.

TSF04

Executes screen space MONTHEND on the 30th of each month at 8:00 AM.

TSF05

Executes and logs screen space STATUS every 30 minutes every day.

Logging parameters for XLF, ASF, and TSF

Exception messages that trip when XLF is in effect are routed to the OMXLFLOG file. Screen space output from ASF and TSF is routed to the OMREPORT file. You can display or change the logging parameters for XLF, ASF, and TSF with the OUTF major command and its minors. For XLF, specify OUTF XLFLOG. For ASF or TSF, specify OUTF REPORT.

In XLF and ASF processing, only one record is written to the log while a given exception condition persists. However, new records will be written to the XLFLOG or the REPORT log if any of the following situations occurs:

- If an exception disappears for even one cycle, and then reappears, it is considered a new event.
- If you turn a given exception off and then on again, you clear the event. If the exception condition still exists, another record goes to the XLF log. OMEGAMON 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 a given 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.

Using screen spaces in ASF and TSF

The power of ASF and TSF lies in their ability to branch to, execute, and log an analysis screen (or series of screens). The screen space you define to call when ASF is activated can contain whatever information-gathering or action-taking commands you specify. You might include commands to turn on the log, change OMEGAMON defaults, further analyze the exception condition, or even call other screen spaces (with .SGO or .FGO).

How ASF and TSF screen spaces execute

If two exceptions occur at the same time, screen spaces and logging execute 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 EXSY appears on any of your target screen spaces, any triggered exception messages appear as usual, but ASF ignores these new exceptions until the sequence is ended with the .RTN command. If the exception condition still exists, ASF goes to the next exception in sequence.

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 terminates. However, when it terminates without a .RTN command, OMEGAMON does not return to the original calling screen, or to another screen defined with .RTN.

You can invoke 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 hung.

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 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.

Using 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 bottom of the last screen space in an ASF or TSF sequence in order to terminate the sequence, return to the original calling screen or branch to the next, and re-enable exception analysis for further automatic calls. 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 .RTN (1 to 9 for numbers 1 to 9 and A to Z for numbers 10 to 35). The following command causes a return to SCREEN2 after 6 cycles.

```
6.RTN SCREEN2
```

On each cycle, OMEGAMON replaces the number in the label field with the next lower number. When the count reaches zero, OMEGAMON fetches SCREEN2.

ASF example

ASF functions only in dedicated mode or in VTAM mode with automatic update in effect. You must set the ASF keyword of the OPTN command to ON to enable the automatic screen feature.

Here are some parameters set with the XACB command for the DRDY exception.

```
XLF Parameters:  
Auto=ON  
Log=OFF  
Limit=3  
Repeat=NO  
Persist=5  
SL=DEX01
```

In this example, when the exception condition exists for 5 cycles in a row, ASF invokes screen space DEX01. From this point on, an A appears in the far right portion of the INFO-line, which indicates that the current screen is part of an ASF sequence.

DEX01 might consist of the following sequence of commands:

```
----- DEX01 DED LOG OM/DEX V420./C SYS 02/12/09 17:03:37 A  
=====
```

> After 8 cycles, branch to screen space DEX02
8.SGO DEX02
=====

> Reset the OMEGAMON cycle time to 15 seconds
.SET INTERVAL=15
=====

> Start degradation analysis
DEX
BEGN
=====

THRS00
PLTS

In this example, the OMEGAMON cycle time is set to 15 seconds. Since 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. At this point, screen space DEX02 is invoked:

```
----- DEX02 DED LOG OM/DEX V420./C SYS 02/12/09 17:03:37 A  
=====
```

> Suspend degradation analysis
CDEG
SUSP
=====

> Return to the calling screen
.RTN

This screen space suspends degradation analysis, and returns to the calling screen with the .RTN command.

Upon return, the ASF sequence terminates (the A disappears from the INFO-line), enabling EXSY for further automatic calls. .RTN also automatically resets the interval to the one in effect when the ASF sequence started. (You could, however, add an NR argument to .RTN (.RTNNR) to direct OMEGAMON to keep the new interval in effect.)

Note that because the SL= parameter was used, ASF automatically turned on the REPORT log when the exception occurred (if it was not already on). When you leave automatic mode, ASF also turns off the log (if it was in the off state before the exception tripped).

When you use ASF to turn on the log automatically with SL=, OMEGAMON first logs the screen in use, and then branches to the scheduled screen space. This is done so that any exceptions can be logged before the ASF sequence begins.

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 [Introduction to troubleshooting](#).

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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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
224A/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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

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

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>.



SC27-4030-02

