Contacting BMC Software

Several methods are available for contacting BMC Software.

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td></td>
<td>or</td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2005-2014 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

CCT and IOS are trademarks or registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

CICS, DB2, IBM, IMS, MQSeries, MVS, OS/390, PR/SM, RACF, RMF, System Storage, System z, VTAM, WebSphere, z/Architecture, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux is the registered trademark of Linus Torvalds.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

UNIX is the registered trademark of The Open Group in the US and other countries.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at [http://www.bmc.com/support](http://www.bmc.com/support). From this website, you can:

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for problems similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report a problem or ask a question
- Subscribe to receive proactive e-mail alerts
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to [customer_support@bmc.com](mailto:customer_support@bmc.com). (In the subject line, enter **SupID:** yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

- **Product information**
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- **Operating system and environment information**
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- **Sequence of events leading to the problem**
- **Commands and options that you used**
- **Messages received (and the time and date that you received them)**
  - Product error messages
  - Messages from the operating system
  - Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

■ Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)

■ In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.

### Chapter 4 Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>55</td>
</tr>
<tr>
<td>APF</td>
<td>56</td>
</tr>
<tr>
<td>ASM</td>
<td>59</td>
</tr>
<tr>
<td>ASVT</td>
<td>64</td>
</tr>
<tr>
<td>AUTHTSO</td>
<td>65</td>
</tr>
<tr>
<td>BBXS</td>
<td>68</td>
</tr>
<tr>
<td>CDE</td>
<td>70</td>
</tr>
<tr>
<td>CLEAR</td>
<td>72</td>
</tr>
<tr>
<td>CNSL</td>
<td>74</td>
</tr>
<tr>
<td>COMMAND</td>
<td>75</td>
</tr>
<tr>
<td>CONSOLES</td>
<td>76</td>
</tr>
<tr>
<td>CPU</td>
<td>77</td>
</tr>
<tr>
<td>CSA</td>
<td>79</td>
</tr>
<tr>
<td>CSMON</td>
<td>82</td>
</tr>
<tr>
<td>CTCB</td>
<td>85</td>
</tr>
<tr>
<td>DEVIATN</td>
<td>87</td>
</tr>
<tr>
<td>DFLIST</td>
<td>90</td>
</tr>
<tr>
<td>DONTSWAP</td>
<td>91</td>
</tr>
<tr>
<td>DSNAME</td>
<td>92</td>
</tr>
<tr>
<td>Command</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>PARMLIST</td>
<td>180</td>
</tr>
<tr>
<td>PBUILD</td>
<td>181</td>
</tr>
<tr>
<td>PFKEY</td>
<td>182</td>
</tr>
<tr>
<td>PIO</td>
<td>183</td>
</tr>
<tr>
<td>PLIST</td>
<td>184</td>
</tr>
<tr>
<td>POST</td>
<td>185</td>
</tr>
<tr>
<td>PPT</td>
<td>186</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>190</td>
</tr>
<tr>
<td>PROGRESS</td>
<td>195</td>
</tr>
<tr>
<td>PRSM</td>
<td>201</td>
</tr>
<tr>
<td>PSTORAGE</td>
<td>203</td>
</tr>
<tr>
<td>QUEUE</td>
<td>208</td>
</tr>
<tr>
<td>REPLIES</td>
<td>214</td>
</tr>
<tr>
<td>RESERVES</td>
<td>215</td>
</tr>
<tr>
<td>RSM</td>
<td>217</td>
</tr>
<tr>
<td>SCT</td>
<td>221</td>
</tr>
<tr>
<td>SDUMP</td>
<td>222</td>
</tr>
<tr>
<td>SETOP</td>
<td>225</td>
</tr>
<tr>
<td>SMFEXIT</td>
<td>226</td>
</tr>
<tr>
<td>SMSLIST</td>
<td>230</td>
</tr>
<tr>
<td>SOFTFRR</td>
<td>233</td>
</tr>
<tr>
<td>SPACE</td>
<td>237</td>
</tr>
<tr>
<td>SRM</td>
<td>242</td>
</tr>
<tr>
<td>SSVT</td>
<td>244</td>
</tr>
<tr>
<td>STATUS</td>
<td>246</td>
</tr>
<tr>
<td>STOKEN</td>
<td>247</td>
</tr>
<tr>
<td>SUBMIT</td>
<td>248</td>
</tr>
<tr>
<td>SVCFIND</td>
<td>249</td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>251</td>
</tr>
<tr>
<td>TCB</td>
<td>252</td>
</tr>
<tr>
<td>TIME</td>
<td>255</td>
</tr>
<tr>
<td>TIOT</td>
<td>256</td>
</tr>
<tr>
<td>TOD</td>
<td>258</td>
</tr>
<tr>
<td>TPIO</td>
<td>260</td>
</tr>
<tr>
<td>TQE</td>
<td>262</td>
</tr>
<tr>
<td>TRACK</td>
<td>264</td>
</tr>
<tr>
<td>TSULIST</td>
<td>267</td>
</tr>
<tr>
<td>UCB</td>
<td>268</td>
</tr>
<tr>
<td>USING</td>
<td>270</td>
</tr>
<tr>
<td>VIO</td>
<td>271</td>
</tr>
<tr>
<td>VMCMD</td>
<td>272</td>
</tr>
</tbody>
</table>
Chapter 5 Exception Monitor samplers

Introduction ................................................................. 281
ACPU ............................................................................. 282
AIO .............................................................................. 283
AIOR ............................................................................ 285
APAG ........................................................................... 287
APGR ........................................................................... 288
ASRV ........................................................................... 290
ASSR ........................................................................... 292
ASVT ........................................................................... 294
BCPU ........................................................................... 295
BIO .............................................................................. 297
BPAG ........................................................................... 298
BSRV ........................................................................... 300
CADP ........................................................................... 302
CHA ............................................................................. 303
CPAG ........................................................................... 304
CPU ............................................................................. 305
CQD ............................................................................. 307
CSA ............................................................................. 307
CSMJ ........................................................................... 309
DEV ............................................................................. 310
DIOQ ........................................................................... 311
DNR ............................................................................. 313
DVTN ........................................................................... 314
ENQ ............................................................................. 316
FIX .............................................................................. 318
INT .............................................................................. 319
JET .............................................................................. 320
JSU .............................................................................. 321
JVR .............................................................................. 323
MTP .............................................................................. 324
NRQ ............................................................................. 325
NRQP .......................................................................... 326
OUT ............................................................................. 328
PAG ............................................................................. 329
| PCPU | ................................................................. | 330 |
| PER  | ........................................................................... | 332 |
| PGQ  | ........................................................................... | 333 |
| PVT  | ........................................................................... | 334 |
| REP  | ........................................................................... | 336 |
| RES  | ........................................................................... | 337 |
| RSU  | ........................................................................... | 338 |
| SMF  | ........................................................................... | 339 |
| SPA  | ........................................................................... | 340 |
| SPLW | ........................................................................... | 342 |
| SRM  | ........................................................................... | 343 |
| SWP  | ........................................................................... | 345 |
| TCPU | ........................................................................... | 346 |
| TIO  | ........................................................................... | 347 |
| TPAG | ........................................................................... | 348 |
| TSRV | ........................................................................... | 350 |
| WDLY | ........................................................................... | 352 |
| WOBJ | ........................................................................... | 353 |
| WRT  | ........................................................................... | 354 |
| WTO  | ........................................................................... | 355 |
| WTO  | ........................................................................... | 356 |
| XCPU | ........................................................................... | 357 |
About this book

This book describes the services offered by the MainView SYSPROG Services product.

It contains information about accessing and invoking SYSPROG Services, and includes detailed information about each service. It is intended for systems programmers, developers, operations personnel, and applications developers.

Throughout this book, references to IBM z/OS support also include support for IBM MVS and OS/390.

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the support website at http://www.bmc.com/support.

Note
Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The MainView SYSPROG Services product includes online Help. In the MainView SYSPROG Services ISPF interface, access Help by pressing PF1 from any ISPF panel.

To access MainView Messages & Codes from any MainView SYSPROG Services panel, type MSG on the COMMAND line.

You can also use the HELP SYSPROG Service. For more information, see “HELP” on page 120.

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/confcenter/index.jsp) to browse documentation sets.
View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:

— Documentation Center

— Support Central (at http://www.bmc.com/support/mainframe-demonstrations)

— BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)

View individual product documents (books and notices) within the “A – Z Supported Product List.”

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.

**Tip**

You can access the BMC Support Central site at http://www.bmc.com/support.

---

**Conventions**

This document uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsys/instance/fileName`

- Menu sequences use a symbol to convey the sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.

---

**Syntax diagrams**

Syntax diagrams are used throughout this book to illustrate syntax usage for services.
Syntax diagrams graphically depict the relationship between literal, variable, and other notational items as understood by an application program or programs. Table 1 on page 13 describes the syntax diagram conventions.

### Table 1: Syntax diagram conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Double right arrowheads" /></td>
<td>Double right arrowheads--the beginning of a syntax diagram.</td>
</tr>
<tr>
<td><img src="image" alt="Single right arrowhead" /></td>
<td>Single right arrowhead--can have two meanings:</td>
</tr>
<tr>
<td></td>
<td>■ When found at the beginning of a line, it indicates that the syntax diagram is continued from another line or diagram.</td>
</tr>
<tr>
<td></td>
<td>■ When found at the end of a line, it indicates that the syntax diagram continues to another line or diagram.</td>
</tr>
<tr>
<td><img src="image" alt="Opposing arrowheads" /></td>
<td>Opposing arrowheads--the end of a syntax diagram.</td>
</tr>
<tr>
<td><img src="image" alt="An item on the main line of the diagram is required" /></td>
<td>An item on the main line of the diagram is required.</td>
</tr>
<tr>
<td><img src="image" alt="A literal is represented in flat text" /></td>
<td>A literal is represented in flat text.</td>
</tr>
<tr>
<td><img src="image" alt="A variable is represented in italic text" /></td>
<td>A variable is represented in italic text.</td>
</tr>
<tr>
<td><img src="image" alt="An item above the main line of the diagram is the default" /></td>
<td>An item above the main line of the diagram is the default.</td>
</tr>
<tr>
<td><img src="image" alt="An item underneath the main line of the diagram is optional" /></td>
<td>An item underneath the main line of the diagram is optional.</td>
</tr>
<tr>
<td><img src="image" alt="Anything under the upward left-branching arrow can be repeated" /></td>
<td>Anything under the upward left-branching arrow can be repeated.</td>
</tr>
<tr>
<td><img src="image" alt="A vertical list of options requires you to choose one" /></td>
<td>A vertical list of options requires you to choose one.</td>
</tr>
<tr>
<td><img src="image" alt="A vertical list of options with the topmost option left blank indicates that you can choose one option or none" /></td>
<td>A vertical list of options with the topmost option left blank indicates that you can choose one option or none.</td>
</tr>
</tbody>
</table>

**MainView SYSPROG Services command syntax example**

The syntax for the CDE service is shown in the following example.
The diagram indicates that the CDE service accepts three positional parameters:

- **asi** – address space instance; see “Identifying address spaces” on page 47 for more information
- **modname** – when specified, displays information about a single module
- **D** – when specified, displays the three-attribute bytes and the address of the CDE in hexadecimal instead of mnemonics for selected attributes

When you invoke a service from the **COMMAND** line, you can use either a blank space or a comma to separate the service name and the parameters, unless you want to omit a parameter, in which case you must use commas to represent the omitted parameter.

**Note**
For clarity, the syntax diagrams in this book do not show commas between parameters.

For example, the following service commands are valid:

- `cde inventory mymod`
- `cde,inventory,mymod`
- `cde @ mymod`
- `cde @,mymod`
- `cde,,mymod`

If you want to issue a command with the default value for a parameter, you do not have to actually type a value for that parameter. For example: CDE @ and CDE are equivalent.

**Using commas as placeholders for default values**

In most cases parameters are positional which means that the position of one parameter relative to any other parameters denotes its purpose. Therefore, if you
choose to omit a parameter that is followed by one (or more) other parameters, you must use commas to represent the omitted parameters.

For example, if you want to display information about a specific module in your own address space (the default), you do not have to type your address space name or even an at sign (@) which represents your address space. However, you must insert commas as placeholders, or enter the omitted parameters, prior to specifying other parameters. For example, the CDE service accepts an address space identifier (asi) as the first parameter and a module name as the second parameter. If you want to refer to your address space, you can omit the second parameter but if you need to specify the second parameter, you must enter two commas to indicate the first parameter is omitted; for example:

cde,,ieavtsdt

**Syntax statements**

This topic explains conventions for showing syntax statements.

A sample statement follows:

| COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
</table>
| Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words. | alias
databaseDirectory
serverHostName |
| Brackets indicate optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option. | [tableName, columnName, field]
[-full, -incremental, -level] |
| Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item. | {DBDName | tableName}
UNLOAD device={disk | tape. fileName | deviceName}
{-a | -c} |
<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A vertical bar means that you can choose only one of the listed items. In the example, you would choose either commit or cancel.</td>
<td>{commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items as many times as necessary.</td>
<td>columnName...</td>
</tr>
</tbody>
</table>
Introducing MainView SYSPROG Services

This section outlines the features, services, and various operation modes of MainView SYSPROG Services.

It concludes with a discussion about the relationship between MainView SYSPROG Services and other BMC Software products.

MainView SYSPROG Services is a real-time software tool that helps your data center manage the day-to-day performance and operation of the operating system environment by helping you to detect, diagnose, and correct problems as they occur, as well as assist you with many important tasks.

The services and Exception Monitor samplers of MainView SYSPROG Services provide the tools for online system management. Data center personnel can delegate tasks to MainView SYSPROG Services by storing a series of operational commands that can be executed automatically.

MainView SYSPROG Services features

MainView SYSPROG Services provides the following features to increase data center productivity, improve system throughput, and manage job deadlines:

- Services
  This feature lets you examine overall system performance, isolate specific system performance delays, and take corrective action. Services are invoked when entered.

- Exception Monitor samplers
  This feature provides early detection of potential performance problems through user-defined threshold indicators. Highlighted warning messages notify you of impending system problems. MainView AutoOPERATOR and other BMC Software products can take actions based on these messages.
SAF security interface

This feature provides an interface to an external security manager (ESM) to prevent unauthorized use of MainView SYSPROG Services.

If the SAF security interface has not been established, MainView SYSPROG Services will disallow all commands. This is true even if an ESM has not been installed.

For information about implementing SAF security for MainView SYSPROG Services, see the following publications:

— MainView Security Guide

Supported environments

MainView SYSPROG Services operates as an authorized program within the MainView Product Address Space (PAS), as a TSO command within a TSO address space, as a batch job, as a Started Task, or from MainView AutoOPERATOR.

You can use MainView SYSPROG Services in the operation modes that best suit your needs. Table 2 on page 18 lists each of the available modes and gives a brief description of each.

Table 2: MainView SYSPROG Services operation modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAS</td>
<td>You can access MainView SYSPROG Services from a MainView User Address Space (UAS). You can also use the MVS MODIFY command to enter MainView SYSPROG commands from a system console.</td>
</tr>
<tr>
<td>TSO</td>
<td>You can invoke MainView SYSPROG Services as a TSO command, in which case it operates in line mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Before you can execute SYSPROG Services as a TSO command, you must add SYSPROG to the AUTHCMD and AUTHPGM lists in the IBM PARMLIB member IKJTSOxx. For more information, see the IBM publication <em>MVS Initialization and Tuning Reference</em>.</td>
</tr>
<tr>
<td></td>
<td>In addition, if SYSPROG Services is running with the MainView for z/OS product, you can use the SYSPROG AUTHTSO service to update the in-storage copies of those lists to enable the use of SYSPROG Services as a TSO command without having to perform a system IPL.</td>
</tr>
<tr>
<td>Started Task</td>
<td>You can also use the MVS MODIFY command to enter MainView SYSPROG Services commands from a system console.</td>
</tr>
</tbody>
</table>
Mode | Description
--- | ---
Batch Job | You can also execute MainView SYSPROG Services in a non-conversational mode, with the input commands obtained from SYSIN and the output directed to SYSPRINT. To operate in this manner, specify PARM=BATCH on the EXEC statement.

For more information about the operation modes, see “Independent operations” on page 44.

Services

Services are executed when you enter them and the results are returned directly to you.

A few of the services collect data over the interval you have requested, in which case the results are returned at the end of the interval. See “Services” on page 55.

**Note**
The LOG and TRACK services run continuously and write the output to a data set.

Exception Monitor samplers

Exception Monitor samplers execute continuously but produce output only when an exception condition is detected, according to the parameters that you specified when you started the Exception Monitor.

The manner in which warning messages are displayed depends upon the environment within which the Exception Monitor is operating. Messages can be retained in a data space for subsequent viewing using the WARN view, or they can be sent to the system console when you are operating under the Product Address Space (PAS).

Messages will be displayed on the system console when the AEW Exception Monitor is being executed as a Started Task.

You can also make exceptions available to the MainView Alert and MainView AutoOPERATOR products for processing. For more information, see the AEWALERT parameter in member “$$INSYS0” on page 24.
How MainView SYSPROG Services works with other products

MainView SYSPROG Services works in conjunction with other BMC Software products, both as a component and as an instrument for gathering data.

The Exception Monitor has the ability to generate Alerts through AOAnywhere, and any MainView Alert or MainView AutoOPERATOR PAS that has ALRTRCVE=YES specified in BBISSP.xx can monitor and respond to various warning messages. Specify AEWALERT=YES in BBPARM member $$INSYS0 to activate this facility. See $$INSYS0 on page 24.
Controlling the MainView SYSPROG Services environment

This section describes the various parameter library members and how you can use them to control various aspects of SYSPROG Services.

How MainView SYSPROG Services locates the parameter library members

When MainView SYSPROG Services is operating under MainView, the various parameter library members are obtained from the set of concatenated partitioned data sets (PDSs) that are allocated to DDNAME PARMLIB. When the product is operating under MainView AutoOPERATOR or in any other environment parameter library, members are obtained from the set of concatenated PDS that is allocated to DDNAME BBIPARM.

Note
To maintain compatibility between releases, if the DDNAMEs BBIPARM and PARMLIB are not associated with any data sets, MainView SYSPROG Services looks for the DDNAME LIB and, if allocated, uses it.

PARMLIST: parameter member processing report

Sometimes it is helpful to have a record of the contents of the various parameter library members that were read during MainView SYSPROG Services initialization and from which specific concatenated data set they were obtained, along with any error messages that were produced.

A report containing this information is written to the data set allocated to DDNAME PARMLIST when it is present. PARMLIST can define a SYSOUT data set or a
preallocated data set with a logical record length of 121 and a record format of FA or FBA.

Using symbols within members

Each MainView SYSPROG Services parameter library member is scanned for symbols that are defined by IBM and, when found, are replaced by the current value of the symbol as they are read. A common use of this facility is to use the IBM variable &SYSCLONE in member $$INSYS0 to define the suffix for other members. For example, specifying AEWSTART=&SYSCLONE facilitates the use of a different list of samplers for each LPAR by using a single parameter library that is shared across multiple LPARS.

Parameter library members

The following table lists each BBPARM library member, along with a brief description and page reference where complete information can be found.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$INEXCL</td>
<td>specifies which job names are excluded from exception reporting by specified Exception Monitor services</td>
<td>“$$INEXCL” on page 23</td>
</tr>
<tr>
<td>$$INLGxx</td>
<td>defines logging system parameters, such as whether the logging output writes to a SYSOUT data set of a specified class or a preallocated data set</td>
<td>“LOGGING” on page 145</td>
</tr>
<tr>
<td>$$INSYS0</td>
<td>defines runtime parameters for MainView SYSPROG Services</td>
<td>“$$INSYS0” on page 24</td>
</tr>
<tr>
<td>$$INTKxx</td>
<td>defines services to track</td>
<td>TRACK on page 264</td>
</tr>
<tr>
<td>$$PVT</td>
<td>specifies limit values for complete or partial address space names for use by the the Exception Monitor's PVT sampler</td>
<td>“$$PVT” on page 28</td>
</tr>
<tr>
<td>$$RPJOBS</td>
<td>exempts jobs with constantly outstanding WTORs from the MainView SYSPROG Services outstanding reply scan</td>
<td>“$$RPJOBS” on page 29</td>
</tr>
<tr>
<td>$$RPTEXT</td>
<td>exempts constantly outstanding WTORs from the MainView SYSPROG Services outstanding reply scan</td>
<td>“$$RPTEXT” on page 31</td>
</tr>
<tr>
<td>$$XDEV</td>
<td>excludes devices from reporting by the Exception Monitor’s DNR sampler</td>
<td>“$$XDEV” on page 30</td>
</tr>
</tbody>
</table>
The following sections describe the BBPARM library members, including syntax rules and examples.

### $$\$\text{INEXCL}$$

This member provides the ability to exclude specific job names from being monitored by these Exception Monitor samplers: ACPU, AIO, AIOR, APAG, APGR, ASRV, ASSR, BCPU, BIO, BPAG, BSRV, CPU, CSMJ, ENQ, JET, JSU, JVR, OUT, REP, RES, RSU, TCPU, TIO, TPAG, TSRV, and WDLY

#### Syntax rules

The format is

```
samplerName jobname1...jobname10
```

The parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>samplerName</code></td>
<td>name of the Exception Monitor sampler that is to exclude selected job names from exception reporting</td>
</tr>
<tr>
<td><code>jobname1-jobname10</code></td>
<td>names of up to ten jobs that are to be excluded from exception reporting. Use as many statements for a sampler as needed to define all job names to be excluded.</td>
</tr>
</tbody>
</table>

#### Defaults

none
Examples

To exclude jobs CPUJOB1, CPUJOB2, and CPUJOB3 from exception reporting by the Exception Monitor CPU sampler, place the following control statement in member $INEXCL:

CPU CPUJOB1,CPUJOB2,CPUJOB3

To eliminate exception reporting for address spaces DUMPSRV and NETS by the OUT service (which monitors swapped-out address spaces), place the following control statement in member $$INEXCL:

OUT NETS,DUMPSRV

$$INSYS0

This member defines runtime parameters for MainView SYSPROG Services.

MainView SYSPROG Services reads this member when it initializes.

Syntax rules

Table 3 on page 25 lists the keyword parameters in member $$INSYS0. The following syntax rules apply:

- Keywords must be entered in the format keyword=value (for example, MSGID=NO).
- Statements can be continued to the next line if they end with a comma.
- Comment lines start with an asterisk.
- Leading blanks prior to the first keyword are allowed.
- Embedded blanks are not allowed; they terminate the scan of the statements.

Note

Route codes are used only when sending unsolicited messages (that is, when the messages are not in response to a request or command).
### Table 3: Keywords for member $SINSYS0$

<table>
<thead>
<tr>
<th>Keyword=value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEWALERT=NO</td>
<td>YES specifies whether Exception Monitor messages should be sent to MainView Alert or MainView AutoOPERATOR PASs that have ALRTRCVE=YES specified in BBISSPxx</td>
</tr>
<tr>
<td>NO--does not send the Exception Monitor messages (default)</td>
<td></td>
</tr>
<tr>
<td>YES--sends the Exception Monitor messages</td>
<td></td>
</tr>
<tr>
<td>AEWNOROLL=0</td>
<td>n specifies the number of Exception Monitor sampler messages not to roll</td>
</tr>
<tr>
<td>0--specifies that all messages roll (default)</td>
<td></td>
</tr>
<tr>
<td>n--specifies that n messages shall not roll, where n is a decimal number from 1 to 10</td>
<td></td>
</tr>
<tr>
<td>Note: The AEWNOROLL value can also be modified by using the SETOP service.</td>
<td></td>
</tr>
<tr>
<td>AEWSTART=00</td>
<td>xx defines the suffix for the default PWSCPMxx member containing control statements for Exception Monitor samplers</td>
</tr>
<tr>
<td>00--uses Exception Monitor control statements in member PWSCPM00 in the BBPARM library (default)</td>
<td></td>
</tr>
<tr>
<td>xx--uses Exception Monitor control statements in member PWSCPMxx in the BBPARM library</td>
<td></td>
</tr>
<tr>
<td>CAPS=OFF</td>
<td>ON specifies how messages should be displayed</td>
</tr>
<tr>
<td>OFF--messages are displayed in mixed case (default)</td>
<td></td>
</tr>
<tr>
<td>ON--messages are displayed in uppercase only</td>
<td></td>
</tr>
</tbody>
</table>
### Keywords and Values

<table>
<thead>
<tr>
<th>Keyword=value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDID=char</td>
<td>defines an MVS subsystem command character. This keyword is valid in Started Task and Batch modes only.</td>
</tr>
<tr>
<td></td>
<td>- char--is a command identification character to route commands to MainView SYSPROG Services using the MVS subsystem interface; valid command identification characters are</td>
</tr>
<tr>
<td></td>
<td>- WTOR--issues a WTOR to prompt for command input when MainView SYSPROG Services is operating as a batch job or started task</td>
</tr>
<tr>
<td>DUMP=NO</td>
<td>specifies whether or not an SVC dump should be created if a service abends</td>
</tr>
<tr>
<td></td>
<td>- NO--does not create an SVC dump if a service abends</td>
</tr>
<tr>
<td></td>
<td>- YES--creates an SVC dump if a service abends (default)</td>
</tr>
<tr>
<td>DVTN=*</td>
<td>specifies where the Exception Monitor's DVTN sampler obtains the CONFIGxx member</td>
</tr>
<tr>
<td></td>
<td>- *--specifies that the DVTN sampler uses the logical PARMLIB concatenation (previously known as SYS1.PARMLIB)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- BBPARM--specifies that the DVTN sampler uses the same set of libraries that were searched to obtain the $$INSYS0 member</td>
</tr>
<tr>
<td></td>
<td>- dsn--specifies that the DVTN sampler uses the specified data set</td>
</tr>
<tr>
<td></td>
<td>This data set is dynamically allocated and freed each time the DVTN sampler executes.</td>
</tr>
</tbody>
</table>

---

Note: When MainView SYSPROG Services are operating within the MainView for z/OS product, you can use its command identification character to send commands to SYSPROG Services by prefixing the command with two command identification characters. For example, if the command identification character is an at sign (@) and you want to execute the CSA,MAP command, enter the following: @@@CSA,MAP
<table>
<thead>
<tr>
<th>Keyword=value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECHO=ECHO</td>
<td>specifies whether MainView SYSPROG input (commands) should be written to the PAS job log</td>
</tr>
<tr>
<td></td>
<td>- ECHO--causes the commands that you enter to be written to the PAS job log (default)</td>
</tr>
<tr>
<td></td>
<td>- NOECHO--causes the commands that you enter <em>not</em> to be written to the PAS job log</td>
</tr>
<tr>
<td></td>
<td>This parameter is applicable only in a PAS environment. You can use the SETOP service to override this value for an individual user.</td>
</tr>
<tr>
<td>EOSMSG=NO</td>
<td>specifies whether MainView SYSPROG Services issues an END OF SERVICE message at the completion of every service</td>
</tr>
<tr>
<td></td>
<td>This keyword is valid in Started Task, TSO, and Batch modes only.</td>
</tr>
<tr>
<td></td>
<td>- NO--does not issue an END OF SERVICE message at the completion of every service (default)</td>
</tr>
<tr>
<td></td>
<td>- YES--issues an END OF SERVICE message at the completion of every service</td>
</tr>
<tr>
<td>RWAIT=nn</td>
<td>sets the global default value for how long (in seconds) the MCOMMAND service will wait for a response</td>
</tr>
<tr>
<td></td>
<td>Each user can use the SETOP service to specify a user-defined RWAIT value. Specify a value between 0 and 15. The default value is 3 seconds.</td>
</tr>
<tr>
<td>LCLAEW=NORMAL</td>
<td>defines where Exception Monitor sampler messages are displayed</td>
</tr>
<tr>
<td></td>
<td>When the Exception Monitor is operating within the MainView PAS, the WARN view is used to view the warning messages on the console when you specify NORMAL. The CONSOLE and BOTH parameters also cause the messages to be displayed on the system console.</td>
</tr>
<tr>
<td></td>
<td>When SYSPROG is executed as a Started Task, the messages are sent only to the system console</td>
</tr>
<tr>
<td>LCLAEWTS=NO</td>
<td>specifies the Exception Monitor sampler messages have a time stamp</td>
</tr>
<tr>
<td></td>
<td>- No- does not time stamp Exception Monitor sampler messages (default)</td>
</tr>
<tr>
<td></td>
<td>- Yes- time stamps Exception Monitor sampler messages</td>
</tr>
</tbody>
</table>
### MSGID=NO

<table>
<thead>
<tr>
<th>Keyword=value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGID=YES</td>
<td>specifies whether message identification numbers prefix MainView SYSPROG Services messages; not valid for TSO mode</td>
</tr>
<tr>
<td>YES--message identification numbers prefix all messages (default)</td>
<td></td>
</tr>
<tr>
<td>NO--no message identification numbers prefix any messages</td>
<td></td>
</tr>
</tbody>
</table>

Message identification numbers appear in tracking output even if you specify MSGID=NO.

### $$PVT$$

This optional member defines threshold values to be used for specific address spaces, or groups of address spaces, whose names begin with the same characters.

It is used by the PVT sampler (see PVT on page 334).

**Note**
The PBUILD service can be used to refresh the in-storage table created from member $$PVT$$ and the PLIST service can be used to display the contents of the in-storage table.

### Syntax rules

The format is:

\[
\text{jobname } \text{limit1 } \text{limit2}
\]

- Statements that begin with an asterisk (*) are considered comments and are ignored.
- Only columns 1 through 72 in each statement are processed.
- The jobname can begin in any column.
- A comma or one (or more) blanks must separate limit1 from the jobname and limit2 from limit1.

The jobname value is required.

- The limit values are optional.
An omitted limit value, or a limit value of zero, indicates that the limit value specified on the PVT sampler control statement should be used.

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **jobname** | one to eight-character jobname of the TSO or batch address spaces  
You must enter the started task with its jobname and started task ID in the form: name.ID. If there is no started task ID, you must repeat the name.  
For example, specify LLA.LLA for the LLA address space. You can specify a partial names by ending the name with an asterisk. For example, to use the same threshold for all of the started tasks with a jobname of ABCDEFGH, specify ABCDEFGH.*.  
MainView SYSPROG Services processes statements based on the the order they are listed in the member. In this example, the member contains the following statements:  
 ABC1 50 50  
 ABC2 50 50  
 ABC* 90,90  
MainView SYSPROG Services uses the threshold value of 50 (percent below-the-line) and 50 (percent above-the-line) for jobs ABC1 and ABC2, and a threshold value of 90 percent for all of the other address spaces whose names begin with ABC. |
| **limit1** | is the limit for below-the-line private storage  
A warning message is issued when the amount of storage is equal to, or greater than, the limit value. If the limit value is zero or is omitted, MainView SYSPROG Services uses the value specified on the PVT sampler statement. |
| **limit2** | is the limit for above-the-line private storage  
A warning message is issued when the amount of storage is equal to, or greater than, the limit value. If the limit value is zero or is omitted, MainView SYSPROG Services uses the value specified on the PVT sampler statement. |

**Note**
No checking is performed for an address space when both limit values are set to 100.

### $$RPJOBS$$

This member exempts jobs with constantly outstanding WTORs that are displayed by the MainView SYSPROG Services REPLIES service and the REP sampler.

See also “$$RPTEXT” on page 31.
**Syntax rules**

The following syntax rules apply:

- Specify one job name per entry.
  
  You can specify a partial job name by using an asterisk (*) at the end of the partial name. For example, ABCD* exempts all of the address spaces that have a job name beginning with ABCD.

- Start each entry in column 1.

- Do not place quotation marks around the job name.

**Default**

none

**Example**

Assume that job IMSDC has a shutdown message outstanding at all times. You can exclude this job from the outstanding reply scan by entering the following statement in member $$RPJOBS:

```
IMSDC
```

**$$XDEV**

This member specifies the addresses of devices that the Device Not Ready (DNR) sampler excludes when reporting devices are not ready.

**Note**

The PBUILD service can be used to refresh the in-storage table created from the member $$XDEV and the PLIST can be used to display the contents of the in-storage table.

**Syntax rules**

- Statements that begin with an asterisk (*) in column 1 are considered comments and are ignored.

- You must enter the device addresses within statement positions 1 through 72 (inclusive). Entries in positions 73 through 80 are ignored.
- The first (or only) device address on a statement can start in any supported position.

- You can specify up to ten device addresses of 3 to 4-character hexadecimal values on each line.

- You can separate device addresses by one or more blanks or, with a comma.

- The maximum number of supported device addresses is 2,032.

**Example**

The following example shows contents of BBPARM member $$XDEV

*The following device addresses will be excluded from reporting by the AEW DNR (Device Not Ready) Sampler.

* 6400 6401 6402 6403 6404 6405 6406 6407 0F50 0F51 F52 F53 F54 F55 F56 F57

---

**$$RPTEXT**

This member exempts constantly outstanding WTORs that could be displayed by the MainView SYSPROG Services REP sampler when they contain specific text.

**Syntax rules**

The following syntax rules apply:

- Specify one entry per statement.

- Place single or double quotation marks around message text containing embedded blanks.

- Enter no more than 24 bytes of message text.

**Default**

none

**Example**

Assume that various jobs might have an outstanding Write To Operator (WTOR) message containing the text **REPLY E TO END**, and you want the REPLIES service and the REP sampler to ignore them. Because the message text contains embedded blanks, you must enclose the text within either apostrophes or quotation marks.
Messages are checked to see if they contain the specified string; therefore, either of the following entries would cause the messages to be excluded:

*REPLY E TO END*

*E TO END*

**XENQ**

This member specifies major/minor name pairs to be excluded from the ENQUEUES and QUEUE service contention displays and from the Exception Monitor ENQ Sampler warning messages.

When a conflict is detected, the exception list is scanned, and if a match is found, the conflict is ignored.

**Syntax rules**

The following syntax rules apply:

- List each major or minor name pair on a single statement.
- Separate the major name from the minor name with a space or a comma.
- If the minor name is omitted, all conflicts with the specified major name are ignored.
- You can specify a partial minor name by terminating the partial name with an asterisk.
- You cannot specify a partial major name.

**Example 1**

All conflicts with a major name of ABC and a minor name of DEF will be excluded.

**ABC,DEF**

**Example 2**

All conflicts with a major name of ABC and a minor name beginning with DEF will be excluded.

**ABC,DEF***
Example 3

All conflicts with a major name of ABC will be excluded, regardless of the minor name.

ABC,*

$$XRES

This member specifies volume serial numbers that are to be excluded from reporting by the RESERVES service and the RES sampler.

Syntax

The following syntax rules apply:

- An asterisk in column 1 indicates that the statement is a comment.
- List one or more complete volume serial numbers within columns 1-71 on a statement, separated by a blank or a comma.
- Use as many statements as desired.

Example

To exclude reserves on volume serial number JESCKP, create BBPARM member $$XRES containing the following statements:

*Exclude the following VOLSERs from reporting by the RESERVES *Service and the RES Sampler
JESCKP

PWSCPMxx

This member contains Exception Monitor sampler control statements.

Syntax rules

The member names have the form PWSCPMxx, where xx is alphanumeric (for example, PWSCPMFF, PWSCPM09, PWSCPM0A9). Each member contains control statements specifying Exception Monitor services, their warning intervals, and appropriate threshold specifications.
The following syntax rules apply:

- Separate the Exception Monitor service name from the specifications with a comma or at least one blank.
- Enter comments by placing an asterisk in column 1.

**Default**

By default, the Exception Monitor Samplers listed in member PWSCPM00 are invoked when you begin Exception Monitor sampling.

**Example**

The following example is of a member called PWSCPMEX:

```plaintext
INT 60
ENQ 5,10
PAG 4,50
AIOR 2,*,250,A
```

The example contains three Exception Monitor sampler statements and a control statement:

- The first statement (**INT 60**) specifies an interval rate of 60 seconds.
- The second statement (**ENQ 5,10**) requests that warning messages be issued at five-minute intervals if any enqueue conflicts have existed for over 10 minutes.
- The third statement (**PAG 4,50**) requests that warning messages be issued at four-minute intervals if the paging rate exceeds 50 over the preceding four minutes.
- The fourth statement (**AIOR 2,*,250,A**) causes warning messages to be issued at two-minute intervals for each address space whose I/O rate exceeds 250 EXCPs per second over the preceding two minutes.
Using MainView SYSPROG Services

This section describes the methods by which you can access MainView SYSPROG Services, either through MainView for z/OS or as a stand-alone product.

MainView SYSPROG Services provides you with the means to control certain internal aspects of your z/OS systems. For example, you can display and alter storage, list the DD names used by a job, add modules and SVCs to the Link Pack Area, and perform a variety of other z/OS system tasks.

If you are running SYSPROG Services through MainView for z/OS, you can access the services in the following ways:

■ From the z/OS and USS Solutions panel, select SYSPROG.
■ From the SYSPROG Services Easy Menu, EZMSPROG, select SYSPROG Fast Menu for an expanded set of options.
■ From the MainView for z/OS easy menu, select SYSPROG Services under the Utilities category.
■ On the COMMAND line, type SYSPROG to display a list of services. You can see additional service help using this method.
■ Use the MODIFY (F) command to invoke a specific service from an operator console.

If you are running SYSPROG Services as a stand-alone product (MainView SYSPROG Services), you can access the services in the following ways:

■ From the z/OS and USS Solutions panel, select SYSPROG.
■ In TSO line mode, type SYSPROG.
■ Use the MODIFY (F) command to invoke a specific service from an operator console.
Accessing MainView SYSPROG Services

You can access MainView SYSPROG Services in different ways.

If you are a MainView for z/OS customer, you can select it through MainView for z/OS. Or, you can access MainView SYSPROG Services directly.

Accessing MainView SYSPROG from the z/OS and USS Solutions panel

The following figure shows the MainView Selection Menu panel.

**Figure 1: MainView Selection Menu panel**

```
--------------------------- MainView Selection Menu ---------------------------
 OPTION    ===>                                                 DATE   -- MM/DD/YY
 TIME       -- HH:MM:SS
0  Parameters and Options                           USERID -- BAOSRR1
E  Alerts and Alarms                                MODE   -- ISPF n.n
P  PLEX Management (PLEXMGR)
U  Utilities, Tools, and Messages
Solutions for
A  Automated Operations
C  CICS
D  DB2
I  IMS
L  Linux
N  Network Management
S  Storage Management
T  Application Management and Performance Tuning
W  WebSphere and MQSeries
Z  z/OS and USS
Enter X to Terminate

Copyright BMC Software, Inc. 2013
```

From this panel, select **Z z/OS and USS**.
The z/OS and USS Solutions panel is displayed, from which you can select **MVzOS** or **SYSPROG**. See Figure 2 on page 37.

**Figure 2: z/OS and USS Solutions panel**

<table>
<thead>
<tr>
<th>OPTION</th>
<th>z/OS and USS Solutions</th>
<th>DATE -- MM/DD/YY</th>
<th>TIME -- HH:MM:SS</th>
<th>USERID -- BAOSRR1</th>
<th>MODE -- ISPF n.n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MVZOS</td>
<td>MainView for z/OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 MVUSS</td>
<td>MainView for UNIX System Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 CMF</td>
<td>CMF MONITOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 SYSPROG</td>
<td>MainView SYSPROG Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Operations**

- 5 CSMON: Common Storage Monitor
- 6 CFM: CMFMON realtime analysis
- H CMFMONHD: CMFMON Historical Data (Available on CMF 5.7 and later)
- 7 CMFUTIL: CMF Extractor Online Utilities
- 8 ANALYZER: Generate CMF Analyzer batch reports
- E ALERTS: Alert Management

**General Services**

- M MESSAGES: Messages and Codes
- P PARMS: Parameters and Options

When you select **4 SYSPROG** from the z/OS and USS Solutions panel, the SYSPROG Easy Menu panel (EZMSPS) is displayed.

To help you access the service that you need, options on the SYSPROG Easy Menu are grouped according to their area of functionality and have been given intuitive, descriptive names. Figure 3 on page 37 shows an example.

**Figure 3: SYSPROG Easy Menu (EZMSPS)**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DATE -- MM/DD/YY</th>
<th>SCREEN MVSPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1 EZMSPS SYSPROG Easy Menu</td>
<td>MVSPS</td>
<td>D1</td>
</tr>
</tbody>
</table>

**Accessing MainView SYSPROG Services from the MainView for z/OS easy menu**

You can also access MainView SYSPROG Services from the MainView for z/OS easy menu (EZMZOS) by selecting the SYSPROG Services option under the Utilities.
category (as shown in the following figure), or by typing **EZMSPROG** on the **COMMAND** line.

**Figure 4: Select MainView SYSPROG Services from the Utilities category**

<table>
<thead>
<tr>
<th>DDMMMYYY</th>
<th>HH:MM:SS</th>
<th>COMMAND</th>
<th>SCROLL</th>
<th>CURR WIN</th>
<th>ALT WIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1 =EZMZOS============CXTSTJ===*========DDMMMYYYY==HH:MM:SS====MVMVS=========1</td>
<td>==&gt;</td>
<td></td>
<td></td>
<td>0026</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity</th>
<th>Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView for z/OS</td>
<td>View RealTime Menu</td>
</tr>
<tr>
<td>Time frame - Interval</td>
<td>View LongTerm Menu</td>
</tr>
</tbody>
</table>

The primary MainView SYSPROG Services menu is the SYSPROG Easy Menu (EZMSPROG), shown in **Figure 3 on page 37**.

### SYSPROG Services Fast Menu

The SYSPROG Services Fast Menu provides more direct access to the services. You can access it by selecting **SYSPROG Fast Menu** from the SYSPROG Easy Menu, or by typing **EZMFPROG** on the **COMMAND** line.

**Figure 5 on page 38** shows the SYSPROG Fast Menu.
Accessing Services from the SYSPROG Services Menu panel

An alphabetical list of all available services is displayed on the SYSPROG Services Menu panel.

To view this panel from MainView for z/OS, go to the MainView for z/OS easy menu panel and type SYSPROG on the COMMAND line. The SYSPROG Services Menu panel is displayed, as shown in Figure 5 on page 38. You can then execute services by typing the command on the COMMAND line or by selecting the service to display its entry panel.

Figure 6: SYSPROG Services Menu panel

<table>
<thead>
<tr>
<th>Valid line commands are:</th>
<th>Valid COMMANDs are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>S - Select service panel</td>
<td>L - Locate a service in the list</td>
</tr>
<tr>
<td>E - Execute service</td>
<td>service - Execute a service</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF</td>
<td>List current APF data sets and volumes</td>
<td></td>
</tr>
<tr>
<td>ASM</td>
<td>Display auxiliary storage manager data</td>
<td></td>
</tr>
<tr>
<td>ASVT</td>
<td>Display address space vector table information</td>
<td></td>
</tr>
<tr>
<td>AUTHTSO</td>
<td>Display/Build TSO lists from IKJTSOxx</td>
<td></td>
</tr>
<tr>
<td>BBXS</td>
<td>Display information for BBX</td>
<td></td>
</tr>
<tr>
<td>CDE</td>
<td>List information about loaded modules</td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td>Clear system dump datasets</td>
<td></td>
</tr>
<tr>
<td>COMMAND</td>
<td>Issue an MVS command from the System Console</td>
<td></td>
</tr>
<tr>
<td>CONSOLES</td>
<td>Display MCS console buffer usage</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Display CPU usage by job</td>
<td></td>
</tr>
<tr>
<td>CSA</td>
<td>Display system usage of csa</td>
<td></td>
</tr>
<tr>
<td>CSMON</td>
<td>Display Common Storage usage by address space</td>
<td></td>
</tr>
<tr>
<td>CTCB</td>
<td>Abnormally terminates the task you select</td>
<td></td>
</tr>
<tr>
<td>DEVIATN</td>
<td>Display configuration deviation</td>
<td></td>
</tr>
<tr>
<td>DONTSWAP</td>
<td>Make an address space non-swappable</td>
<td></td>
</tr>
<tr>
<td>DSNNAME</td>
<td>Display dataset attributes and volumes</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>Display the contents of main storage</td>
<td></td>
</tr>
<tr>
<td>EDITINFO</td>
<td>Display EDT information</td>
<td></td>
</tr>
<tr>
<td>ENQUEUEES</td>
<td>Displays enqueue conflicts</td>
<td></td>
</tr>
</tbody>
</table>

Scroll down to display the rest of the services. Each service is described in Services on page 55.

Performing functions

You can perform the following actions from the SYSPROG Services Menu panel list:

- Locate a service.
- Find out more about a service.
- Execute a service.

Table 4 on page 40 summarizes the functions of the MainView SYSPROG Services commands L, S, and E; subsequent sections in this chapter provide more details.
Table 4: Available MainView SYSPROG Services commands

<table>
<thead>
<tr>
<th>To</th>
<th>Do this action</th>
<th>For more information, see</th>
</tr>
</thead>
<tbody>
<tr>
<td>search for a service by name</td>
<td>Type <code>L serviceName</code>.</td>
<td>“Locating a service” on page 40</td>
</tr>
<tr>
<td>find out more about a service, or change its parameters</td>
<td>Use the <code>S</code> line command.</td>
<td>“Finding out more about a service” on page 40</td>
</tr>
<tr>
<td>execute a service</td>
<td>Choose one:</td>
<td>“Executing a service” on page 41</td>
</tr>
<tr>
<td></td>
<td>■ Use the <code>E</code> line command (uses defaults).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ On the <code>COMMAND</code> line, type the name of the service and its parameters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Use the <code>S</code> line command to select the panel so that you can specify the parameters, and then press <code>Enter</code>.</td>
<td></td>
</tr>
</tbody>
</table>

Locating a service

To locate a service—for example, LNKLST—type `L LNKLST` on the COMMAND line. LNKLST is scrolled to the top of the display.

Finding out more about a service

To find out more information about a service, place the cursor next to the desired service, type an `S` (Select), and press `Enter`. The appropriate Service Information panel is displayed, describing the service and providing input fields in which you can specify parameters.
Figure 7 on page 41 shows the Service Information panel for the LNKLST service.

Figure 7: Service Information panel for LNKLST

--- SYSPROG LNKLST Service ---

COMMAND ===> More: +
ACTION ===> _________ (PROCESS, ACTIVATE, UPDATE, DEACT,
UNDEFINE, STATUS, LIST, USERS or
LOCATE)
SETNAME ===> ________________ (Use with ACTIVATE, DEACT, LIST,
UNDEFINE and USERS)
SUFFIXES ===> __ ===> __ (PROGxx suffixes. Use with PROCESS)
JOBNAME ===> ________ (Use with UPDATE)
MODULE ===> ________ (Use with LOCATE)

Press ENTER to EXECUTE SERVICE
END to CANCEL REQUEST

PROCESS - Process the LNKLST DEFINE, LNKLST ADD, LNKLST DELETE,
and LNKLST UNDEFINE statements in the PROGxx
member(s). Used to create and modify LNKLST sets.

Executing a service

To execute a service from the SYSPROG Services Menu, perform one of the following actions:

- On the COMMAND line, type the name of the service and its accompanying parameters.
- Type E (Execute) to the left of the service and press Enter.
- Type S (Select) to the left of the service and press Enter. Specify the parameters on
  the Service Information panel, and press Enter.

The output for that service appears in the SYSPROG Services Output panel.

Figure 8 on page 41 shows the SYSPROG Services Output panel after the LNKLST service is executed.

Figure 8: SYSPROG Services Output panel for LNKLST

--- SYSPROG Services Output Log ---

Command ===> LNKLST,LIST
SCROLL ===> CSR
TARGET - MVCR61E

14:00:35 CMD=LNKLST,LIST
AMTK2FI Seq APF SMS Volser Data Set Name
AMTK2GI --- --- --- ----- --------------------------
The SYSPROG Services Output panel displays the output for all services, with the most recently executed service positioned at the bottom of the display. You can scroll up or down within the panel and invoke other services by typing the appropriate name and parameters on the COMMAND line.

### Executing MainView SYSPROG Services from an operator console

There are three execution environments supported by MainView SYSPROG Services that either require or permit communication through an IBM MVS operator console.

### Operating as a Started Task or a batch job

There are three options in this environment:

- Use the MVS MODIFY command to enter input (commands).
  
  This is the default. The syntax for using the MVS MODIFY command to execute MainView SYSPROG Services commands is:

  \[ F \text{name},\text{command} \]

  — \text{name} is the jobname or the started task ID for the MainView SYSPROG Services address space

  — \text{command} is the MainView SYSPROG Services command that you want to execute
• Define a command ID character and use it to prefix the command.
  The syntax for using a command ID character to execute a MainView SYSPROG Services command is:
  
  `?command`
  
  — `?` is the command ID character that you defined
  — `command` is the MainView SYSPROG Services command that you want to execute

• Specify that MainView SYSPROG Services issues a WTOR that prompts you to provide input.
  To use option 2 or 3, you must specify either a command character or WTOR with the CMDID parameter in member $$INSYS. For more information, see “$INSYS0” on page 24.

**Operating under MainView *for z/OS***

In this environment, you can always enter commands with the MVS MODIFY (F) command and, if a command ID character has been defined, with the command ID character.

However, in this environment, you must define the command ID character with the CMDID parameter in the PAS JCL (and not with the CMDID parameter in member $$INSYS0).

The syntax for using the MVS MODIFY command to execute a MainView SYSPROG Services commands is:

`F pasname,R=command`

The syntax for using a command ID character to execute a MainView SYSPROG Services command is:

`??,command`

— `??` is the command ID character

— `command` is the MainView SYSPROG Services command that you want to execute
Note
In this environment, two command ID characters are required. The first routes the request to the PAS and the second routes the request to MainView SYSPROG Services.

---

**Operating under MainView AutoOPERATOR or the General Invocation Interface (GII)**

In this environment, MainView SYSPROG Services ignores the CMDID parameter and always issues a WTOR to prompt for input.

---

**Valid CMDID characters**

The following characters are valid CMDID characters.

They can be specified as a single-character or a hexadecimal equivalent (two digits) in the form 'Xcc'.

ε . < ( + | & * ) - / % _ > ? : @ ' = "

Note
Many of these characters must be specified in hexadecimal format to be accepted by the parsing routine.

---

**Independent operations**

MainView SYSPROG Services has the ability to operate independently from MainView in the following modes:

- Started Task (STC)
- TSO (Line)
- batch job
Executing MainView SYSPROG Services as a Started Task

When MainView SYSPROG Services is executed as a Started Task (STC), it receives commands from an operator console and sends the response to that console.

Commands can be entered using the MVS Modify Command, a command ID character, or by responding to an outstanding WTOR prompt. For more information about the CMDID parameter, see “$$INSYS0” on page 24.

To use MainView SYSPROG Services as a started task, refer to the instructions and sample JCL in BBSAMPLIB member SYSPROGJ.

To start MainView SYSPROG Services as a started task, type

S SYSPROG.id

where id is the (optional) started task identifier.

Note

It is possible to start MainView SYSPROG Services prior to starting the Job Entry Subsystem (JES). However, if you do, all referenced cataloged data sets must be cataloged in the Master catalog. In addition, SYSOUT data sets cannot be used because they require JES services.

Executing Services in Started Task mode

If you are running MainView SYSPROG Services in Started Task mode, you can enter commands by using a command ID character.

For example, assuming that the greater than symbol (>) is the user-defined command ID character, to invoke the CPU service, type

>CPU

For more information about user-defined command ID characters and other options, see Table 3 on page 25.

A console must have SYS or ALL authority before you can execute commands from it. The message CONSOLE NOT AUTHORIZED is displayed if the authority has not been assigned.

After you to enter a command, MainView SYSPROG Services responds with the following message:

AMT00DI Enter SYSPROG command (sysid)
MainView SYSPROG Services as a TSO command

To invoke MainView SYSPROG Services in TSO Line mode, you can issue one of the following commands:

- At the TSO READY prompt, type SYSPROG.
- On an ISPF COMMAND line, type TSO SYSPROG.
- From ISPF Option 6, type SYSPROG.

The following message is displayed:

AMTIN1I SYSPROG SERVICES RELEASE v.r.mm INITIALIZED
AMT001A SYSPROG

You can now issue MainView SYSPROG Services commands. If you need to see a list of the commands, type HELP at the AMT001A SYSPROG prompt on the COMMAND line.

To display the last service used, type ? at the AMT001A SYSPROG prompt.

To repeat the last service, type *.

To terminate the MainView SYSPROG Services command, type END at the AMT001A SYSPROG prompt.

**Note**

To access MainView SYSPROG Services in TSO line mode, you need to have ‘hilevel.BMCPSWD’ in your LOGON proc or the linklist.

**WARNING**

Before you can execute SYSPROG Services as a TSO command, you must add SYSPROG to the AUTHCMD and AUTHPGM lists in the IBM PARMLIB member IKJTSOxx. For more information, see the IBM publication *MVS Initialization and Tuning Reference*.

In addition, if SYSPROG Services is operating under the MainView for z/OS product, you can use the SYSPROG AUTHTSO service to update the in-storage copies of those lists and enable the use of SYSPROG Services as a TSO command without having to perform a system IPL.

MainView SYSPROG Services as a batch job

You can execute a series of services as a batch job.
Output is placed in the SYSPRINT data set.

To do so, place MainView SYSPROG Services commands in the JCL shown in Figure 9 on page 47 and submit the job for processing.

**Figure 9: Sample batch job for MainView SYSPROG Services**

```plaintext
//SYSPROG    EXEC  PGM=SYSPROG,REGION=1024K,PARM='BATCH'
//STEPLIB    DD   DSN=hilevel.BBLINK,DISP=SHR
//BMCPSWD    DD   DISP=SHR,DSN=?????????.BMCPSWD
//BBPARM     DD   DSN=hilevel.BBPARM,DISP=SHR
//SYSPRINT   DD   SYSOUT=A
//SYSIN      DD   *
CSA         [1]
ASM,MAP     [1]
END         [2]
/*
```

**Legend**

1. CSA and ASM,MAP are the MainView SYSPROG Services commands and parameters to be executed.

2. Specify **END** as the last statement (optional).

**Identifying address spaces**

Many services display information for, or perform an action within, a specified address space.

The descriptions of those services use the term asi (address space instance) to identify the target address space. You can identify the target address space several ways, by

- "at sign" character (@) to reference your address space
- asterisk (*) to indicate the previously referenced address space
- **jobname, started task ID, fully-qualified name** or TSO **USERID**
- **ASID** expressed as a decimal value by enclosing it in parentheses, such as (123)
- **ASID** expressed as a hexadecimal value by preceding the value with an X and enclosing it in parentheses; for example, (X123), which is equivalent to (291)
- **job instance token (JIT)**

Note that work on a system is dynamic; jobs continuously terminate and start in the same address space and occasionally the jobs use the same name. Also, the operating
system supports concurrent execution of multiple started tasks with the same job name but with different started task identifiers. Therefore, it is important to ensure that you are referencing the correct job instance.

To make this task easier, MainView SYSPROG Services supports the use of a started task ID and most services accept a fully-qualified name. A fully-qualified name consists of a jobname and a started task ID (separated by a period) in the form `jobName.startedTaskID`. You can also enter only the jobname or the started task ID. You must prefix the started task ID with a period (.), which indicates that this is an ID and not a jobname. For example, `JES2` refers to the started task with an ID of JES2.

To ensure that you are referring to the same job instance each time you execute a service, you can use the JITOKEN service to obtain a unique token for the job instance and then use the token it provides in place of the jobname on subsequent services.

The MainView SYSPROG Services JITOKEN service, if given any of the elements in the above list, returns a token that uniquely identifies the job instance. Using a job instance token ensures that a requested action will not be taken against a different job instance with the same job name or ASID. After the job instance terminates, the token is invalid and will be rejected by any service.

When you use an asterisk (*) to indicate the last-referenced address space, the job instance token of the last-referenced address space is substituted. When services are invoked from a row representing an address space in a MainView view, the job instance token is used to identify the target address space.

For more information, see JITOKEN on page 131.

**Expressions**

Several MainView SYSPROG Services accept address expressions for one or more parameters.

To achieve maximum benefits from the product, you need to understand how expressions are used. An expression evaluates to a target address and can optionally denote the length of the data at that address. Expressions are evaluated in the context of the target address space.

Address expressions consist of one or more terms, separated by an operator. The first term can be a previously defined symbol. Each term can be suffixed with one or more indirection indicators. The expression `ASCB+6C?+C0` contains three terms: `ASCB`, `6C?`, and `C0`, which are separated by the plus sign operator. (The minus sign is also supported.)
In the example ASCB+6C?+C0

- **ASCB** is a predefined symbol that represents the address of the ASCB control block for the target address space.
  
  For more information about symbols, see “Symbols” on page 50. MainView SYSPROG Services currently provides approximately 150 predefined symbols. You can also use the EQUATE service to define your own symbols. For more information, see EQUATE on page 109.

- **6C?** is a term consisting of a hexadecimal value (6C), followed by an indirection indicator (?).
  
  For more information, see “Indirection indicators” on page 49.

- **C0** (a hexadecimal value) is another term. You can also express values in decimal by appending an N to the decimal value. For example, 192N has the same value as C0. Additional numeric scaling suffixes are also provided. See “Scale suffixes” on page 51.

- If the first term is a symbol, the target address space attribute of the expression is inherited from that symbol. The exception is if the expression is explicitly overridden by specifying another target address space name with another parameter on the command.

In the example DUMP,XYZ+8,20, MainView SYSPROG Services targets the address space implied by symbol XYZ.

In the example DUMP,XYZ+8,20,ABC, MainView SYSPROG Services targets the address space ABC because ABC is an explicit reference.

For more information, see EQUATE on page 109.

### Indirection indicators

When the expression processor encounters an indirection indicator, it converts the preceding part of the expression to an address and then uses that address to reference storage and obtain a replacement address (in accordance with the type of indirection symbol).

The four types of indirection symbols and their attributes are as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>A percent sign indicates that the full-word value at the indicated address is to be treated as a 24-bit address value. In other words, the value of the high-order byte is ignored, and the remaining three bytes are substituted for the address.</td>
</tr>
</tbody>
</table>
Symbols must consist of one-to-eight characters and begin with an alphabetic character (A-Z) or one of these special characters: @, #, or $.

Numerals (0-9) can be used in the second and subsequent character positions.

Note
You should try to avoid creating symbols that might be confused with hexadecimal values (for example, symbols that contain only the characters A through F and 0 through 9.

Symbols are replaced by their definition as they are encountered, and then the result is processed. Therefore, the definition of a symbol can (and frequently does) contain another symbol. For example, the definition of the predefined symbol ASXB is ASCB +6C?, the definition of ASCB is PSAAOLD?, and the definition of PSAAOLD is 224. Therefore, the symbol ASXB will be converted to 224?+6C? by the expression processor. The address at PSAAOLD is the address of the target address space's ASCB.

Last-Referenced Storage Pointer
Several of the MainView SYSPROG Services set the contents of the Last-Referenced Storage Pointer (LRSP) to the beginning of the area where it is located in storage.

You can use this value in any service that accepts an expression by coding an asterisk as the first (or only) term in the expression.
For example, the DUMP service stores the beginning address of each storage area that it displays in the LRSP. Therefore, you can easily display the contents of a series of linked control blocks. Assume that you have already displayed the first control block and now want to follow the chain, and that the second word of the control block contains a full-word pointer to the next control block. In this case, "DUMP *+4? will display the next block in the chain. Subsequent blocks on the chain can be displayed by repeating the command. To simplify things further, you can repeat the last command by just entering an asterisk (*). Thus, to display the third and subsequent blocks, all you need to enter is an asterisk.

It is also convenient to use the LRSP when using the ZAP service to change the contents of a storage area that you have just displayed.

**Target address space**

As previously stated, expressions are evaluated in the context of the target address space.

Most MainView SYSPROG Services that accept an expression also permit you to explicitly specify the target address space. However, you might not need to do so because the target address space is implied by the first evaluated symbol.

---

**Note**

Most predefined symbols have a null address space attribute. In that case, your address space will be the implied (default) address space.

---

In addition to a value, each symbol has a length attribute and a target address space attribute. The attributes that are associated with the first evaluated symbol in an expression are retained and used when you do not specify a length or target address space.

For example, the OLIST service creates a symbol name for each memory object in the specified address space. You can subsequently refer to the memory object by its symbolic name; the address space that contains it will be implied. You can also use the EQUATE service to associate an address space with a symbol that you define.

**Scale suffixes**

Large values are scaled to fit within the width of the display field by displaying the most-significant digits, followed by a suffix that denotes the actual magnitude.
For example the value 4096 might be displayed as 4K. Several services also accept scaled values as input. For example, the WARNING service permits the use of scaled values when changing the values for an Exception Monitor Sampler.

Table 5 on page 52 lists the suffix characters:

### Table 5: Suffix characters

<table>
<thead>
<tr>
<th>Suffix character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>indicates that the preceding decimal value is in Kilobytes and should be scaled by multiplying it by 10 to the 3rd power</td>
</tr>
<tr>
<td>M</td>
<td>indicates that the preceding decimal value is in Megabytes and should be scaled by multiplying it by 10 to the 6th power</td>
</tr>
<tr>
<td>G</td>
<td>indicates that the preceding decimal value is in Gigabytes and should be scaled by multiplying it by 10 to the 9th power</td>
</tr>
<tr>
<td>T</td>
<td>indicates that the preceding decimal value is in Terabytes and should be scaled by multiplying it by 10 to the 12th power</td>
</tr>
<tr>
<td>P</td>
<td>indicates that the preceding decimal value is in Petabytes and should be scaled by multiplying it by 10 to the 15th power</td>
</tr>
<tr>
<td>Ki</td>
<td>indicates that the preceding decimal value is in Kilobytes and should be scaled by multiplying it by 2 to the 10th power</td>
</tr>
<tr>
<td>Mi</td>
<td>indicates that the preceding decimal value is in Megabytes and should be scaled by multiplying it by 2 to the 20th power</td>
</tr>
<tr>
<td>Gi</td>
<td>indicates that the preceding decimal value is in Gigabytes and should be scaled by multiplying it by 2 to the 30th power</td>
</tr>
<tr>
<td>Ti</td>
<td>indicates that the preceding decimal value is in Terabytes and should be scaled by multiplying it by 2 to the 40th power</td>
</tr>
<tr>
<td>Pi</td>
<td>indicates that the preceding decimal value is in Petabytes and should be scaled by multiplying it by 2 to the 50th power</td>
</tr>
</tbody>
</table>

**Working with 64-bit addresses**

MainView SYSPROG Services provides some syntax shortcuts to facilitate entering 64-bit addresses.

First, you can optionally use an underscore character to separate the high-order and low-order words of a 64-bit address. If you do so, you can also omit all leading zeros on each word. For example, the value 0000020000000000 can also be entered as

- 00000200_00000000
- 200_0
Specifying the start and length of a comparison string

With some services (for example, OFIND), you can specify a search string. Generally, you will want to enter these values as literals. However, it is sometimes easier to refer to a value that is currently in storage. You can do this reference by using an expression to define the start of the string and then add the length of the string, enclosed in parentheses, to the end of the expression; for example: expression_value(length). See OLIST on page 176.
Services

The services offered by MainView SYSPROG Services help improve and increase productivity by allowing you to manipulate various z/OS internals.

This section provides detailed information about services, including proper syntax, required and optional parameters, and usage examples. For information about using the Services Menu, see Using MainView SYSPROG Services on page 35.

ANALYZE

The ANALYZE (ANA) service examines the system for possible performance-related issues and reports any that it discovers.

Syntax

```
ANALYZE
```

Example

To detect possible performance-related issues (such as a backlog of work, or jobs waiting for an enqueue), type

**ANALYZE**

The ANALYZE service detected a backlog of work waiting to be dispatched.

The following results indicate a backlog of work:

```
AMTAN5I 3 Address spaces are waiting to be dispatched
```

The following results indicate that jobs are waiting for an enqueue:

```
AMTAN1I 2 waiting for resource held by VAM3NQ1 (0061) SJSD?AMTAN2I QNAME=PIONON,RNAME=Test_ENQ_Rname
```
APF

This service lists the current authorized program libraries and their volume serial numbers.

This service also:

■ selectively lists libraries by data set name, partial data set name, or volume serial number

■ adds a new entry to the APF list; authorizes link list data sets that were not authorized at IPL

■ deletes an entry from the APF list

■ changes the volume serial number for an entry in the APF list

■ provides ability to change the volume serial number for an entry in the APF list

■ provides ability to selectively modify the link list after adding or deleting an entry from the APF list

■ provides option to omit the volser on any command; unlike the SETPROG command and the CSVAPF macro, which require the specification of a volser, the APF service scans the catalog for the correct volser if one is not specified

Syntax

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsname</td>
<td>specifies the data set name of the authorized library</td>
</tr>
<tr>
<td>partialDsname</td>
<td>represents the first characters of a data set name</td>
</tr>
<tr>
<td></td>
<td>For instance, if you type APF RXA15, the service lists every data set that</td>
</tr>
<tr>
<td></td>
<td>has a prefix of RXA15.</td>
</tr>
<tr>
<td>volser</td>
<td>specifies the volume serial number</td>
</tr>
<tr>
<td></td>
<td>■ If a volser is not specified, the cataloged volser for the data set is</td>
</tr>
<tr>
<td></td>
<td>used.</td>
</tr>
<tr>
<td></td>
<td>■ If <em>SMS</em> is specified as the volser, SMS will manage the data set.</td>
</tr>
<tr>
<td></td>
<td>■ If ****** is specified as the volser, the system residence volume is used.</td>
</tr>
<tr>
<td>ADD</td>
<td>adds the specified data set on the specified volume to the APF list</td>
</tr>
<tr>
<td>DELETE</td>
<td>deletes the specified data set from the APF list</td>
</tr>
<tr>
<td>CHANGE</td>
<td>changes the volser on which the data set resides to the volume that follows</td>
</tr>
<tr>
<td></td>
<td>■ If a volser is not specified, the cataloged volser for the data set name</td>
</tr>
<tr>
<td></td>
<td>is used.</td>
</tr>
<tr>
<td></td>
<td>■ If <em>SMS</em> is specified as the volser, SMS will manage the data set.</td>
</tr>
<tr>
<td></td>
<td>■ If ****** is specified as the volser, the system residence volume is used.</td>
</tr>
</tbody>
</table>

**Example 1**

To list all entries in the APF list, type

APF

```
AMTA71I   APF LIST AT LOCATION 01B08000 HAS 201 ENTRIES
AMTA72I   VOLUME OPNS00    DSN RES1.RES31.BBLINK
```

Legend:

1. hexadecimal address of the APF list in storage
2. APF list
3. volser of the authorized library, *SMS*, or ******
4. data set name of the authorized library

**Note**

The APF service repeats message AMTA72I for every entry in the APF list.
Example 2

To list all APF entries that start with RES1, type

*APF RES1*

*AMTA771* THE FOLLOWING ENTRIES MATCHED THE REQUEST
*AMTA721* VOLUME OPNS00 DSN RES1.RES31.BBLINK

Example 3

To list all APF entries for any data set on volser OPNS00, type

*APF,OPNS00*

*AMTA721* VOLUME OPNS00 DSN RES1.RES30.BBLINK

**Note**
The two commas indicate omission of the first parameter.

Example 4

To create a new entry in the APF list for RES1.RES30.BBLOAD, residing on OPNS01, type

*APF RES1.RES30.BBLOAD,OPNS01,ADD*

*AMTA731* APF LIST NOW CONTAINS ENTRY FOR
*AMTA721* VOLUME OPNS01 DSN RES1.RES30.BBLOAD

Example 5

To create a new entry in the APF list for a cataloged data set, type

*APF RES1.RES30.BBLOAD,ADD*

*AMTA731* APF LIST NOW CONTAINS ENTRY FOR
*AMTA721* VOLUME OPNS01 DSN RES1.RES30.BBLOAD

The APF service determines where RES1.RES30.BBLOAD is cataloged and creates a new APF list with the entered data set and the cataloged volser.

Example 6

To add a data set to the APF list and have the data set managed by SMS, type

*APF RES1.USER.FILE,*SMS*,ADD*

*AMTA731* APF LIST NOW CONTAINS ENTRY FOR
*AMTA721* VOLUME *SMS* DSN RES1.USER.FILE
*AMTA7MI* SMS MANAGED DATASET RES1.USER.FILE

Example 7

To remove an entry from the APF list, type
Example 8

To change the volser of an existing APF entry, type

**APF RES1.RES30.BBLOAD,OPNS01,CHANGE,PERF01**

AMTA751 APF LIST ENTRY CHANGED. OLD VOLUME SER OPNS01 IS NOW
AMTA72I VOLUME PERF01    DSN RES1.RES31.BBLOAD

ASM

The ASM service displays information for unusual Auxiliary Storage Manager (ASM) conditions and the ASM paging data sets as well as the auxiliary page data set slots that are currently in use by each address space.

Syntax

```
ASM
   STATUS
   DSN
   ASID|MAP
   NAME
   VIO
   limit
   TOTAL
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATUS</td>
<td>displays the following information for each page data set: data set sequence number, device number, number of extents, data set type (PLPA, Common, or Local), total number of slots, and the current percentage of available (free) slots. The second parameter is ignored if specified. See example 1.</td>
</tr>
<tr>
<td>DSN</td>
<td>displays the following information for each page data set: data set sequence number, data set name, data set type (PLPA, Common, or Local), total number of slots, and the current percentage of available (free) slots. The second parameter is ignored if specified. See example 2.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ASID      | displays information sorted in ascending ASID sequence  
All address spaces are displayed unless limited to \textit{nnn} by the second parameter. |
| MAP       | is an alias for ASID to maintain compatibility with prior releases |
| NAME      | sorts information in ascending jobname sequence  
All address spaces are displayed unless limited to \textit{nnn} by the second parameter. |
| TOTAL     | sorts information in descending TOTAL sequence (sum of non-VIO and VIO slots in use by the address space)  
The display is limited to the top ten address spaces unless \textit{nnn} is specified as the second parameter, in which case the top \textit{nnn} address spaces are displayed. |
| NVIO      | sorts information in descending NVIO sequence (number of non-VIO slots in use by the address space)  
The display is limited to the top ten address spaces unless \textit{nnn} is specified as the second parameter, in which case the top \textit{nnn} address spaces are displayed. |
| VIO       | sorts information in descending VIO sequence (number of VIO slots in use by the address space)  
The display is limited to the top ten address spaces unless \textit{nnn} is specified as the second parameter, in which case the top \textit{nnn} address spaces are displayed. |
| limit     | requests that all address spaces owning at least limit number of non-VIO slots be displayed  
The output is sorted in descending order by non-VIO slots. |

The following parameters are valid only for the second parameter position and in addition, the following conditions apply:

- The second parameter is ignored when the first parameter is STATUS, DSN, or limit.
- The default value for the second parameter is 10 when the first parameter is VIO, NVIO, or TOTAL.
- The default value is ALL when the first parameter is ASID or NAME.

\textit{nnn} limits the display to the first \textit{nnn} address spaces

\textit{ALL} indicates that the information for all address spaces is to be displayed

### Column headers

**NUM**

The data set number assigned by the operating system. The first is the PLPA data set, the second is the Common page data set, and the remaining are Local page data sets.
DVN

The four-character hexadecimal device number.

DSN

The data set name.

VOLSER

The volume serial number of the volume containing the page data set.

#Ex

The number of extents allocated for the page data set.

E

VIO eligibility status. Y indicates the data set is eligible for VIO use and N indicates it is not eligible for VIO use.

Note: Message AMTA5BW is displayed if VIO has spilled into ineligible data sets.

Type

The type of paging data set. PLPA for PLPA, Comm for Common, Local for Local pages data sets, and SCM for Storage Class Memory.

Slots

The total number of allocated slots (pages) available in the data set.

Pct Free

The percentage of the allocated slots that are currently available for use.

Jobname

The name of the job currently executing in the address space.

ASID

The Address Space Identifier (ASID) for the address space.

Total

The total number of local page data set slots in use by the address space (the sum of the Non-VIO and VIO slots in use).
Non-VIO

The number of non-VIO slots in use by address space.

VIO

The number of VIO slots in use by the address space.

Pct Used

The percentage of slots used by this address space to the total number of slots.

Example 1

To display the type of IPL, any unusual ASM-related conditions and the device number, VOLSER, number of extents and other information for each page data set, type

ASM

Warning messages are displayed when an unusual condition is present.

```
AMTA5SCI Last IPL was Cold Start (CLPA)
AMTA52I Total local slots= 40,268K Available= 39,323K ( 97.65%)
AMTA5DI Largest holder of slots is MTADMSTR with 0.22%
AMTA5DI
AMTA5HI NUM     DVN   VOLSER  #Ex  E  Type       Slots   Pct Free
AMTA5II ---     ----  ------  ---  -  -------  --------  --------
AMTA53I   0     85B3  SADPG1    1  N  PLPA      17,999    14.27%
AMTA53I   1     85B3  SADPG1    1  N  Comm      53,999    99.88%
AMTA53I   3     85B3  SADPG1    1  Y  Local       135K  100.00%
AMTA53I   4     85B3  SADPG1    1  Y  Local       135K  100.00%
AMTA53I   5     A205  SADPG2    1  Y  Local       180K  100.00%
AMTA53I   6     4A10  SADPG3    1  Y  Local       180K  100.00%
AMTA53I   7     A205  SADPG2    1  Y  Local       180K  100.00%
AMTA53I   8     A205  SADPG2    1  Y  Local       180K  100.00%
AMTA53I   9     85B3  SADPG1    1  Y  Local      53,999   100.00%
AMTA53I  10     4A10  SADPG3    1  Y  Local       180K  100.00%
AMTA53I  11     4A10  SADPG3    1  Y  Local       180K  100.00%
AMTA53I  12     4A1F  SADPG4    1  Y  Local       180K  100.00%
AMTA53I  13     4A1F  SADPG4    1  Y  Local       180K  100.00%
AMTA53I  14     4A1F  SADPG4    1  Y  Local       180K  100.00%
AMTA53I  15     495A  SADPG5    1  Y  Local       594K  100.00%
AMTA53I  16     85E5  SADPG6    1  Y  Local       594K  100.00%
AMTA53I  17     A202  SADPG7    1  Y  Local       594K  100.00%
AMTA53I  18     495E  SADPG8    1  Y  Local       594K  100.00%
AMTA53I  19     C06  SADPG9    1  Y  Local       594K  100.00%
AMTA53I  20     AD74  SADPGA    1  Y  Local       600K  100.00%
AMTA53I  21     AD74  SADPGA    1  Y  Local       600K  100.00%
AMTA53I  22     AD74  SADPGA    1  Y  Local       600K  100.00%
AMTA53I ** N/A *SCM*     0  N  SCM      33,554K  97.18%
```

Example 2

To display the type of IPL, any unusual ASM-related conditions and the data set name, and other information for each page data set, type

ASM, DSN

```
AMTA5SCI Last IPL was Cold Start (CLPA)
AMTA52I Total local slots= 40,268K Available= 39,317K ( 97.64%)
AMTA5DI Largest holder of slots is MTADMSTR with 0.23%
```

62 MainView SYSPROG Services User Guide and Reference
Example 3

To display the top-10 consumers of local page data set slots, type

```
ASM,TOTAL
```

<table>
<thead>
<tr>
<th>Jobname</th>
<th>ASID</th>
<th>Total</th>
<th>Non-VIO</th>
<th>VIO</th>
<th>Pct Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS70BSS</td>
<td>0111</td>
<td>91,574</td>
<td>91,574</td>
<td>0</td>
<td>2.11</td>
</tr>
<tr>
<td>WS61BSS</td>
<td>00DB</td>
<td>87,954</td>
<td>87,954</td>
<td>0</td>
<td>2.03</td>
</tr>
<tr>
<td>WS70BSA</td>
<td>0110</td>
<td>80,210</td>
<td>80,210</td>
<td>0</td>
<td>1.85</td>
</tr>
<tr>
<td>WS70BS</td>
<td>00F7</td>
<td>67,600</td>
<td>67,600</td>
<td>0</td>
<td>1.56</td>
</tr>
<tr>
<td>OMVS</td>
<td>000F</td>
<td>62,599</td>
<td>62,599</td>
<td>0</td>
<td>1.44</td>
</tr>
<tr>
<td>WS61BS</td>
<td>00FF</td>
<td>59,014</td>
<td>59,014</td>
<td>0</td>
<td>1.36</td>
</tr>
<tr>
<td>TRACE</td>
<td>0004</td>
<td>49,341</td>
<td>49,341</td>
<td>0</td>
<td>1.14</td>
</tr>
<tr>
<td>DIB1DBM1</td>
<td>0041</td>
<td>33,931</td>
<td>33,931</td>
<td>0</td>
<td>0.78</td>
</tr>
<tr>
<td>MTQ7MSTR</td>
<td>07E</td>
<td>30,730</td>
<td>30,730</td>
<td>0</td>
<td>0.71</td>
</tr>
<tr>
<td>SYSPROG</td>
<td>00AF</td>
<td>28,342</td>
<td>28,342</td>
<td>0</td>
<td>0.65</td>
</tr>
</tbody>
</table>

If none of the address spaces are using any local page data set slots, the following message is displayed:

```
There are no address spaces holding local slots
```

Example 4

To display the top-20 users of non-VIO local page data set slots, type

```
ASM,NVIO,20
```

<table>
<thead>
<tr>
<th>Jobname</th>
<th>ASID</th>
<th>Total</th>
<th>Non-VIO</th>
<th>VIO</th>
<th>Pct Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS70BSS</td>
<td>0111</td>
<td>91,574</td>
<td>91,574</td>
<td>0</td>
<td>2.11</td>
</tr>
<tr>
<td>WS61BSS</td>
<td>00DB</td>
<td>87,954</td>
<td>87,954</td>
<td>0</td>
<td>2.03</td>
</tr>
<tr>
<td>WS70BSA</td>
<td>0110</td>
<td>80,210</td>
<td>80,210</td>
<td>0</td>
<td>1.85</td>
</tr>
<tr>
<td>WS70BS</td>
<td>00F7</td>
<td>67,600</td>
<td>67,600</td>
<td>0</td>
<td>1.56</td>
</tr>
<tr>
<td>OMVS</td>
<td>000F</td>
<td>62,599</td>
<td>62,599</td>
<td>0</td>
<td>1.44</td>
</tr>
<tr>
<td>WS61BS</td>
<td>00FF</td>
<td>59,014</td>
<td>59,014</td>
<td>0</td>
<td>1.36</td>
</tr>
<tr>
<td>TRACE</td>
<td>0004</td>
<td>49,341</td>
<td>49,341</td>
<td>0</td>
<td>1.14</td>
</tr>
<tr>
<td>DIB1DBM1</td>
<td>0041</td>
<td>33,931</td>
<td>33,931</td>
<td>0</td>
<td>0.78</td>
</tr>
<tr>
<td>MTQ7MSTR</td>
<td>07E</td>
<td>30,730</td>
<td>30,730</td>
<td>0</td>
<td>0.71</td>
</tr>
<tr>
<td>SYSPROG</td>
<td>00AF</td>
<td>28,342</td>
<td>28,342</td>
<td>0</td>
<td>0.65</td>
</tr>
</tbody>
</table>
Example 5

To display all address spaces holding at least 10,000 slots type either of the following:

```
ASM,10000

ASM 10K
```

<table>
<thead>
<tr>
<th>AMTA54I</th>
<th>Jobname</th>
<th>ASID</th>
<th>Total</th>
<th>Non-VIO</th>
<th>VIO</th>
<th>Pct Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTA54J</td>
<td>MTADMSTR</td>
<td>0134</td>
<td>91,394</td>
<td>91,394</td>
<td>0</td>
<td>0.23%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MT9MSTR</td>
<td>0135</td>
<td>86,467</td>
<td>86,467</td>
<td>0</td>
<td>0.21%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>OMVS</td>
<td>000F</td>
<td>52,194</td>
<td>52,194</td>
<td>0</td>
<td>0.13%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DJP1IBM1</td>
<td>00DA</td>
<td>45,226</td>
<td>45,226</td>
<td>0</td>
<td>0.11%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>JS70SS</td>
<td>018D</td>
<td>43,317</td>
<td>43,317</td>
<td>0</td>
<td>0.11%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DEHRB1M1</td>
<td>00EB</td>
<td>37,438</td>
<td>37,438</td>
<td>0</td>
<td>0.09%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTADBCD</td>
<td>012B</td>
<td>27,781</td>
<td>27,781</td>
<td>0</td>
<td>0.07%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DJJ2DBM1</td>
<td>0131</td>
<td>26,548</td>
<td>26,548</td>
<td>0</td>
<td>0.07%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>RMF</td>
<td>002F</td>
<td>24,233</td>
<td>24,233</td>
<td>0</td>
<td>0.06%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>BCVDV1G</td>
<td>0192</td>
<td>19,024</td>
<td>19,024</td>
<td>0</td>
<td>0.05%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTAMSTR</td>
<td>0136</td>
<td>16,767</td>
<td>16,767</td>
<td>0</td>
<td>0.04%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>RASP</td>
<td>0003</td>
<td>16,705</td>
<td>16,705</td>
<td>0</td>
<td>0.04%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DJQ2DBM1</td>
<td>012D</td>
<td>16,468</td>
<td>16,468</td>
<td>0</td>
<td>0.04%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>JS70SA</td>
<td>018A</td>
<td>14,824</td>
<td>14,824</td>
<td>0</td>
<td>0.04%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>ZFS</td>
<td>0045</td>
<td>14,472</td>
<td>14,472</td>
<td>0</td>
<td>0.04%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTAGMSTR</td>
<td>0133</td>
<td>14,089</td>
<td>14,089</td>
<td>0</td>
<td>0.03%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>M7PMSSTR</td>
<td>003E</td>
<td>12,934</td>
<td>12,934</td>
<td>0</td>
<td>0.03%</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTA4NG2D</td>
<td>0129</td>
<td>11,018</td>
<td>11,018</td>
<td>0</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

The ASVT (ASV) service displays information for three address space queues maintained in the ASVT control block:

- available queue
- non-reusable replacement queue
- Start/SASI queue

The following information is displayed for each queue:

- current number of slots (ASIDs) on the queue
percentage of original number of slots (ASIDs) on the queue

original number of slots (ASIDs) on the queue

name of system parameter that defined the original queue length

When the system marks an ASID as nonreusable, it replaces the ASID by transferring a slot from the nonreusable replacement queue to the available queue until the replacement queue is empty. If an attempt is made to start a Started Task and the available queue is empty, an ASID on the Start/SASI queue is used.

**Syntax**

```
ASTV
```

**Example**

To display information for the three address space queues, type

```
ASVT
```

<table>
<thead>
<tr>
<th>ASID</th>
<th>Description</th>
<th>Current</th>
<th>Percent</th>
<th>Original</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTAV1</td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTAV2</td>
<td>AVAILABLE QUEUE</td>
<td>225</td>
<td>45.00</td>
<td>500</td>
<td>MAXUSER</td>
</tr>
<tr>
<td>AMTAV3</td>
<td>NON-REUSABLE REPLACEMENT QUEUE</td>
<td>43</td>
<td>28.66</td>
<td>150</td>
<td>RSVNONR</td>
</tr>
<tr>
<td>AMTAV31</td>
<td>START/SASI QUEUE</td>
<td>25</td>
<td>100.00</td>
<td>25</td>
<td>RSVSTRT</td>
</tr>
<tr>
<td>AMTAV4</td>
<td>NUMBER OF NON-REUSABLE ASID</td>
<td>107</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AUTHTSO**

Use the AUTHTSO service to modify the contents of the TSO authorization lists for the duration of the current IPL. This service eliminates the need to IPL and lets you test updates before entering them into production.

The AUTHTSO (AUT) service:

- lists the contents of the current authorization lists
- builds a new authorization list from any IKJTSO.xx member in SYS1.PARMLIB
- restores the authorization list built at IPL

Use the AUTHTSO service to modify the contents of the TSO authorization lists for the duration of the current IPL. This service eliminates the need to IPL and lets you test updates before entering them into production.
TSO authorization lists define which TSO commands or programs can be invoked with APF authorization. AUTHTSO accesses each TSO authorization list independently of the other lists. The three types of TSO authorization lists are as follows:

- **AUTHCMD (CMD)**—authorized TSO commands
- **AUTHPGM (PGM)**—authorized TSO programs
- **AUTHTSF (TSF)**—APF-authorized programs that can be called through the TSO Service Facility (TSF)

**Syntax**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>lists the contents of the current authorization list</td>
</tr>
<tr>
<td>BUILD</td>
<td>builds a new authorization list from any IKJTSOxx member in SYS1.PARMLIB; 00 is the default</td>
</tr>
<tr>
<td>RESTORE</td>
<td>restores the authorization list built at the time of IPL</td>
</tr>
</tbody>
</table>
### Parameters and Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>applies the appropriate command to all three lists: TSO commands, programs, and Service Facility</td>
</tr>
<tr>
<td>CMD</td>
<td>specifies authorized TSO commands</td>
</tr>
<tr>
<td></td>
<td>You can substitute AUTHCMD, IKJEFT2, or E2 for CMD.</td>
</tr>
<tr>
<td>PGM</td>
<td>specifies authorized TSO programs</td>
</tr>
<tr>
<td></td>
<td>You can substitute AUTHPGM, IKJEFT2, or E2 for PGM.</td>
</tr>
<tr>
<td>TSF</td>
<td>specifies the TSO Service Facility</td>
</tr>
<tr>
<td></td>
<td>You can substitute AUTHTSF, IKJEFTAP, or AP for TSF.</td>
</tr>
<tr>
<td>xx</td>
<td>suffix of SYS1.PARMLIB member IKJTSOxx</td>
</tr>
</tbody>
</table>

### Examples

The following examples illustrate the ways to invoke the AUTHTSO service.

#### List examples

To display the authorization list of TSO commands (CMD), type

**AUTHTSO LIST CMD**

```
[1] AMTAT7I AUTHORIZED COMMAND LIST (IKJEFT2) -
[2] AMTAT7I BUILT DURING IPL AT 02:16 ON 13.215:
[3] AMTATZI
[4] AMTAT8I RECEIVE TRANSMIT XMIT LISTD
[5] AMTAT8I SEND RACONVRT SYNC SYSPROG
```

Legend:

1. type of authorization list (COMMAND, PROGRAM, or APF-TSF)
2. process by which the authorization list was built (IPL or AUTHTSO)
3. time and date the authorization list was created
4. commands in the specific authorization list

To display all authorization lists, type

**AUTHTSO**

```
AMTAT7I AUTHORIZED COMMAND LIST (IKJEFT2) -
AMTAT7I BUILT DURING IPL AT 02:16 ON 13.215:
AMTATZI
AMTAT8I RECEIVE TRANSMIT XMIT LISTD
AMTAT8I SEND RACONVRT SYNC SYSPROG
AMTATZI
AMTAT7I AUTHORIZED PROGRAM LIST (IKJEFTER) -
AMTAT7I BUILT DURING IPL AT 02:16 ON 13.215:
AMTATZI
AMTAT8I IEBCOPY SPFCOPY IKJEFT76 SYSPROG
AMTAT8I ICHUT100 ICHUT200 ICHUT400
AMTATZI
```
Build example

Refer to the appropriate IBM manual for syntax rules for the IKJTSOxx member.

To build a new authorization list of TSO commands from SYS1.PARMLIB member IKJTSO01, type

```
AUTHTSO BUILD IKJEFTE2 01
```

```
AMTAT9A REPLY Y TO CONFIRM "BUILD" OF TSO CMD LIST, N TO CANCEL
```

```
AMTAT8I TSO CMD LIST (IKJEFTE2) BUILT
```

 Restore example

To restore all authorization lists to their status at IPL, type

```
AUTHTSO RESTORE ALL
```

```
AMTAT9A REPLY Y TO CONFIRM RESTORE OF TSO CMD LIST, N TO CANCEL
```

```
AMTAT8I TSO CMD LIST (IKJEFTE2) RESTORED
```

```
AMTAT9A REPLY Y TO CONFIRM RESTORE OF TSO PGM LIST, N TO CANCEL
```

```
AMTAT8I TSO PGM LIST (IKJEFTE8) RESTORED
```

```
AMTAT9A REPLY Y TO CONFIRM RESTORE OF TSO TSF LIST, N TO CANCEL
```

```
AMTAT8I TSO TSF LIST (IKJEFTAP) RESTORED
```

Usage notes

- The AUTHTSO service prompts you for a response before it updates each of the three authorization lists. To restore the original authorization list, type Y in response to message AMTAT9A.

- To use the original authorization list, you must log off and then log back on to TSO after AUTHTSO restores the list. TSO creates a static copy of the authorization list in your TSO address space when you log on.

BBXS

The BBXS (BBX) service displays the status of the BMC Software subsystem services.

When diagnosing problems, BMC Software Support staff might ask you to use the BBXS service to verify information about the subsystem.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBXS</td>
<td>displays information about the BBXS subsystem (default)</td>
</tr>
<tr>
<td>subsystemID</td>
<td>displays information about the specified BMC Software subsystem service (when it is named something other than BBXS)</td>
</tr>
<tr>
<td>ALL</td>
<td>displays a summary line for each BBXS subsystem ID</td>
</tr>
</tbody>
</table>

Example 1

To display information about the BBXS subsystem, type

BBXS

```
AMTBXAD BBX Subsystem Information
AMTBXDD
AMTBX2D Subsystem name=BBXS; SSCT=00BD3270 ## AUTOINIT ##
AMTBX3D BBCT addr=00B5A000; restart count=01; RMID=BPB1160 , LEVEL=0005
AMTBX5D Initialized by ITSTNPAS (mod=CX10SDVR,key=4) at 03:49:32 on 08/03/13
AMTBX6D BBX was loaded from BB.ITSTN.BBLINK
AMTBX4D Previous BBCT Addr=00B9F000; RMID=BPB1144 , level=0005 ## AUTOINIT ##
AMTBX5D Initialized by DC$BBI (mod=ASTXAMN,key=8) at 03:47:48 on 08/03/13
AMTBX6D BBX was loaded from SYSO.$BBPROD.BBLINK
```

Legend:

1 name of the BMC Software subsystem service currently running
2 address of the Subsystem Services Control Table (SSCT)
3 address of the BMC Software Subsystem Services Control Table (BBCT)
4 number of times that the subsystem was restarted
5 RMID of the subsystem
6 name of the address space that created the subsystem
7 program that created the subsystem
8 TCB key of the address space that created the subsystem
9 time that the subsystem was created
10 date that the subsystem was created
11 data set from which the subsystem modules were loaded
12 information about the previously loaded BBCT

**Example 2**

To display summary information for each BBXS subsystem, type

```
BBXS ALL
```

The CDE service lists all of the modules or a specific module that is loaded in a specified address space, and displays information about them.

**Syntax**

```
CDE  @  asi  modname
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>specifies your own address space</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see Identifying address spaces on page 47 for more information</td>
</tr>
<tr>
<td>modname</td>
<td>limits the display to the specified module</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>displays the three-attribute bytes and the address of the CDE in hexadecimal instead of mnemonics for selected attributes</td>
</tr>
</tbody>
</table>

### Examples

There are two forms of output, controlled by the presence or absence of a D in the third parameter position. When D is specified, the hexadecimal contents of the three attributes bytes are displayed rather than decoding the attribute values, and the address of the CDE is displayed.

#### Example 1 (without D)

To display information about modules in address space PCAUTH, type

**CDE,PCAUTH**

```
AMTY40I Address Space: PCAUTH
AMTY41I AMTY42I TCB address is 5FEB1B
AMTY43I Listing of associated CDE's
AMTY45I Name    A Load Adr Length Entry Pt Loc Usect Attributes
AMTY46I IEAVAR00 081D31C8 23F8 B10D31C8 EFLPA 269 RENT AUTH
AMTY44I
AMTY42I TCB address is 5FDE28
AMTY43I Listing of associated CDE's
AMTY45I Name    A Load Adr Length Entry Pt Loc Usect Attributes
AMTY46I IEAVXMAS 1AF00740 18C0 9AF01328 252 1 RENT AUTH
AMTY46I IEEPRWI2 00DC2000 10E0 80DC2A40 PLPA ----- RENT AUTH
AMTY46I IEAVXPCA 1AF020B0 CF50 9AF020B0 252 1 RENT AUTH
```

#### Example 2 (with D)

To display information about modules in address space PCAUTH, type

**CDE,PCAUTH,D**

```
AMTY40I Address Space: PCAUTH
AMTY41I
AMTY42I TCB address is 5FEB1B
AMTY43I Listing of associated CDE's
AMTY45I Name    A Load Adr Length Entry Pt Loc Usect Attr Bytes CDE Adrs
AMTY46I IEAVAR00 081D31C8 23F8 B10D31C8 EFLPA 269 b1 22 00 00FBC0A8
AMTY44I
AMTY42I TCB address is 5FDE28
AMTY43I Listing of associated CDE's
AMTY45I Name    A Load Adr Length Entry Pt Loc Usect Attr Bytes CDE Adrs
AMTY46I IEAVXMAS 1AF00740 18C0 9AF01328 252 1 31 22 00 00FBC0A8
AMTY46I IEEPRWI2 00DC2000 10E0 80DC2A40 PLPA ----- B1 22 00 00B77D7B
AMTY46I IEAVXPCA 1AF020B0 CF50 9AF020B0 252 1 31 22 00 00FBC0A8
```
Legend:

1 name

2 column header; an asterisk in the A column indicates that the name to its left is an alias

3 module entry point address

4 module use count

5 list of primary attribute flags that are on

6 address of the Contents Directory Entry (CDE)

CLEAR

The CLEAR (CLE) service clears system dump data sets without printing the information.

Use the CLEAR service in conjunction with the SDUMP service (which takes an SVC dump of one or more address spaces) and the SYSDUMP service (which lists the SVC dump data sets). See “SDUMP” on page 222 and “SYSDUMP” on page 251.

Note

System-generated dump data set names are not supported by this service.

Syntax

The parameters are defined as follows:
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>clears all system dump data sets (default)</td>
</tr>
<tr>
<td>id</td>
<td>displays a one- or two-digit number representing the system dump data set to be cleared</td>
</tr>
<tr>
<td>PROMPT</td>
<td>lists the title of the dump contained in each full dump data set processed; in addition, the system issues a confirmation message</td>
</tr>
</tbody>
</table>

Valid responses to the confirmation message are as follows:

- Y clears the system dump data set
- E terminates the CLEAR service

If you type any character other than Y or E, the current system dump data set will be bypassed and the next full system dump data set (if any) will be processed.

### Example

To clear the system dump data set #01, type

```
CLEAR 01
```

```
AMTC50I SYS1.DUMP01 WAS FILLED AT 10:20 ON JAN 21,2013[1]
AMTC51I SOURCE=SVCDUMP [2]
AMTC52I TITLE=LOW CORE OVERLAY (LOOP) [3]
AMTC50A ENTER Y TO CLEAR SYS1.DUMP01 [4]
 y
AMTC53I 'SYS1.DUMP00' DATA SET CLEARED [5]
AMTC5AI CLEAR      PROCESSING COMPLETE
```

**Legend:**

1. when the specified dump was taken
2. system component that took the dump
3. title of the dump
4. operator verification is required before the specified data set is cleared
5. dump data set has been successfully cleared

### Usage notes

- You can omit the first operand without using delimiting commas.
- You can abbreviate the operands this way:
  - A for ALL
  - N for NOPROMPT
CNSL

The CNSL (CNS) service simulates an operator console at a TSO user’s terminal so that you can use MCS console facilities to resolve problems.

**Note**
This service is only available when SYSPROG Services is executing as a TSO command.

### Syntax

```
CNSL
```

### Example

To simulate an operator console, type

```
CNSL
```

To simulate an operator console, type

```
CNSL
```

Legend:

1. address of the console that is currently selected; press **PF6** (ALTCONS) to select an alternate console

2. control settings that are currently in effect for the selected MCS console

---

MainView SYSPROG Services User Guide and Reference
3 number of action messages that are currently displayed

4 number of replies that are currently outstanding for this console

5 number of intervention-required messages that are currently outstanding for this console

Usage notes

■ The CNSL service selects an active MCS console for simulation and displays it in the same format at the TSO terminal.

■ The CNSL service is not updated automatically as the display at the MCS console is updated. Press Enter or a PF key to update the CNSL service display.

■ You can enter any operator command. Press PF5 to redisplay the last command entered.

■ Functions of the MVS control (K) command that affect the formatting of the referenced Console are not permitted by the CNSL service.

■ The MainView SYSPROG Services terminal must have an 80-column screen.

■ If the MainView SYSPROG Services terminal has fewer lines than the MCS console selected for simulation, the most recent lines are displayed.

■ Press PF3 (END) to exit.

COMMAND

The COMMAND (COM) service provides the ability to execute any system or JES2 command as if it were entered from a system console if the user has the authority to execute the command.

Syntax

```
COMMAND command
```

The parameter is defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>operator command</td>
</tr>
</tbody>
</table>

**Example 1**

To use the COMMAND service to enter the operator command to stop Dump Analysis and Elimination (DAE), type

`COM,SET DAE=10`

**Example 2**

To execute the command `D U,TAPE`, type

`COM, D U,TAPE`

**JES2 command example**

To execute the display active (D A) JES2 command, type

`COM,D A`

**CONSOLES**

The CONSOLES (CON) service displays the number of undisplayed messages per system console and the number of reply buffers in use.

This service helps you determine which console in an MCS environment is not operational.
With this service, you can also determine the number of console reply buffers in use when diagnosing the cause of a wait state (for example, when a job that must issue a WTOR is waiting for reply buffers to be freed by other tasks in the system).

Syntax

```
CONSOLES
```

Example

To list the number of messages to be displayed, type

```
CONSOLES
```

```
AMTC32I   CONSOLE 0A0 HAS   1 MESSAGES TO BE DISPLAYED
AMTC32I   CONSOLE 0A2 HAS 198 MESSAGES TO BE DISPLAYED
AMTC33I   198 OF 200 CONSOLE BUFFERS IN USE
AMTC34I   1 OF 006 REPLY BUFFERS IN USE
```

Legend:

1 console address
2 number of messages to be displayed; a high number indicates a message backlog
3 total number of console buffers in use
   This number can be less than the sum of messages to be displayed at the various
   consoles because a message can be routed to more than one console.
4 number of reply buffers in use

CPU

The CPU service provides information about job activity by noting the proportionate use of CPU resources by job.

Use CPU to monitor system activity.
Syntax

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{time}</td>
<td>length of the sample period in seconds; the default is 10 seconds, and the maximum value is 1,200 seconds</td>
</tr>
</tbody>
</table>

Example

To display information about CPU usage for the past 10 seconds, type

\texttt{CPU, 10}

\begin{verbatim}
AMTOB7I Collecting CPU data for 10 seconds
AMTOB11 Plex Busy 90.2% [1]
AMTOB11 Plex Wait 6.0% [2]
AMTOB11 Plex Overhead 3.8% [3]
AMTOB11 Partition Rcvd. 2.3% which is [4]
AMTOB11 52.2% of its Relative Share (of 4.4%) [5]
AMTOB01 AMTOB01
AMTOB31 Used Priority Part. Rcvd.
[6] [7] [8] [9]
AMTOB41 STC WLM 8.6% FF (255) .38%
AMTOB41 STC RMFGAT 6.4% EB (232) .28%
AMTOB41 STC C500M 5.0% EB (232) .22%
AMTOB41 STC SONCAS 2.1% EB (232) .09%
AMTOB41 STC XCFAS 1.9% FF (255) .08%
AMTOB41 TSO BMHTR3 1.9% FF (255) .08%
AMTOB41 STC C600M 1.7% EB (232) .07%
AMTOB41 STC GRS 1.0% FF (255) .04%
[10]
AMTOB51 All others used 23.6% 1.03%
[11]
AMTOB61 TOTAL USED 52.2% 2.30%
AMTOB01
AMTOB21 Batch= 1.0%, STC= 95.1%, TSO= 3.9%, Total=100.0% [12]
AMTOB01
AMTOB11 Sample begin time 11:50:41.596
AMTOB11 Sample end time 11:50:51.597
\end{verbatim}

Legend:

1 percentage of the sample period that the sysplex was performing useful work

2 percentage of the sample period that the processors were not dispatched to a partition because the partitions were waiting

3 percentage of the sample period used by the hardware to dispatch processors to partitions
percentage of the sample period that the partition hosting the SYSPROG Services session was dispatched

Subtracting this percentage from the PLEX BUSY percentage yields the percentage of the sample that all other partitions in the sysplex were dispatched.

percentage of the partition’s relative share (the amount of processing time that the partition is supposed to receive when the partition’s demands are high and resources are limited) used during the sample period

When the partition can use more time and time is available, it can exceed 100 percent of its relative share. Likewise, the partition can (when wait assist is off) receive less than its relative share.

type of address space: a batch job (JOB), Started Task (STC), or TSO session (TSO)

names of the address spaces that consumed the highest percentage of the total CPU used by the LPAR during the sampling period, listed in descending order of CPU utilization

Address spaces that use less than one percent are not listed. Up to ten address spaces are listed.

CPU usage by address space

priority of the address space

CPU usage for all other address spaces in the LPAR

sum of CPU usage for all address spaces in the partition during sample interval; sum is equal to the partition’s relative share usage (56.8 percent in this example)

usage distribution among batch jobs, Started Tasks, and TSO sessions

The CSA service displays the utilization of:

- Common Service Area (CSA)
- Extended Common Service Area (ECSA)
- System Queue Area (SQA)
- Extended System Queue Area (ESQA)
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM</td>
<td>provides a summary of common storage (CSA and SQA) usage</td>
</tr>
<tr>
<td>MAP</td>
<td>provides a detailed display of CSA/ECSA usage by subpool and storage key</td>
</tr>
<tr>
<td>BOTH</td>
<td>provides both the summary and the detailed displays</td>
</tr>
</tbody>
</table>

Example 1

To display a detailed description of CSA and ECSA usage, type

**CSA,MAP**

<table>
<thead>
<tr>
<th>AMTCMAI</th>
<th>Owner</th>
<th>SUP</th>
<th>SCHED</th>
<th>VSPC</th>
<th>AVM</th>
<th>DM</th>
<th>VTAM</th>
<th>IMS</th>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTMCBI</td>
<td>SP/Key</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>227</td>
<td>17K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24K</td>
<td>21K</td>
<td>73K</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>228</td>
<td>233K</td>
<td>3K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43K</td>
<td>0</td>
<td>406K</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>231</td>
<td>1K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>73K</td>
<td>48K</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMCI</td>
<td>Non-Extended CSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>227</td>
<td>150K</td>
<td>147K</td>
<td>144</td>
<td>198K</td>
<td>4K</td>
<td>4M</td>
<td>1M</td>
<td>5M</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>228</td>
<td>10M</td>
<td>13K</td>
<td>215K</td>
<td>0</td>
<td>44K</td>
<td>2M</td>
<td>2M</td>
<td>67M</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>231</td>
<td>2M</td>
<td>11K</td>
<td>0</td>
<td>13K</td>
<td>979K</td>
<td>2M</td>
<td>6M</td>
<td>59M</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMCI</td>
<td>Extended CSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI</td>
<td>241</td>
<td>87M</td>
<td>1M</td>
<td>2M</td>
<td>58K</td>
<td>25M</td>
<td>1M</td>
<td>1M</td>
<td>173M</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTCMEI</td>
<td>Total</td>
<td>0</td>
<td>99M</td>
<td>1M</td>
<td>2M</td>
<td>72K</td>
<td>70M</td>
<td>5M</td>
<td>22M</td>
<td>200M</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1. This line and the line above it represent the column labels:
   - **SP** stands for Subpool (first column).
   - Key describes the rest of the columns, which are key numbers.
   - Above each key is a mnemonic describing the system component that allocates storage in that key.
The subpool number appears first, followed by the amount of storage, if any, allocated in that key within the subpool.

3 total amount of storage allocated with that key, and the sum allocated for all keys

Note
User-key CSA (keys 8-15) are not displayed and not included in the totals.

Example 2

To display an overview of CSA usage, type

CSA

Legend:

1 total amount of CSA/ECSA defined

2 amount of space converted to SQA/ESQA and percentage of total CSA/ECSA

3 total amount of space currently used and percentage of total CSA/ECSA
The CSMON service displays information about common storage and the address space to which its ownership is attributed.

Note
The CSMON Service provides a subset of the information provided by the CSMON facility that is included in the MainView for z/OS product.

You can use short-term monitoring to diagnose sudden increases in common storage allocations, or to monitor one or more address spaces that could be overusing common storage.

You can also use long-term monitoring to diagnose slowly increasing allocations of common storage, or to determine whether specific address spaces are allocating excessive amounts of common storage. For more information about common storage, commands, data extraction and analyses, see the COMMON STORAGE MONITOR User Guide.

Syntax

```
CSMON
```

The parameters are defined as follows:
### Example 1

To list the common storage blocks allocated by JES2, enter:

**CSMON,MAP JES2**

<table>
<thead>
<tr>
<th>ASID</th>
<th>START - END</th>
<th>LENGTH</th>
<th>SP</th>
<th>KY</th>
<th>EYEBALL</th>
<th>DATE</th>
<th>TIME</th>
<th>ASID</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTCSI</td>
<td>00B4B198-00B4B21F</td>
<td>136</td>
<td>227</td>
<td><em>.</em>//.<em>,</em>.. *</td>
<td>2013/09/18 22:21:19</td>
<td>001D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00BA8300-00BAAF6F</td>
<td>11376</td>
<td>241</td>
<td><em>MIT HASC</em></td>
<td>2013/09/18 22:14:30</td>
<td>001D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00BB74C0-00BB74FF</td>
<td>64</td>
<td>228</td>
<td>*$ECBU... *</td>
<td>2013/09/18 22:14:31</td>
<td>001D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>summarizes CSA and SQA allocations, both above and below the 16 MB line, by address space. The default value is SUMMARY for CSMON. The following list describes the parameters that you can use with SUMMARY:</td>
</tr>
<tr>
<td>ASID</td>
<td>sorts the output by ASID column. This is the default.</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>sorts the output by the JOBNAME column.</td>
</tr>
<tr>
<td>SQA</td>
<td>sorts the output by the TOT SQA column in descending order.</td>
</tr>
<tr>
<td>SQA&lt;</td>
<td>Sorts the output by the SQA&lt; column in descending order.</td>
</tr>
<tr>
<td>SQA&gt;</td>
<td>Sorts the output by the SQA&gt; column in descending order.</td>
</tr>
<tr>
<td>CSA</td>
<td>sorts the output by the TOTCSA column in descending order.</td>
</tr>
<tr>
<td>CSA&lt;</td>
<td>Sorts the output by the CSA&lt; column in descending order.</td>
</tr>
<tr>
<td>CSA&gt;</td>
<td>Sorts the output by the CSA&gt; column in descending order.</td>
</tr>
<tr>
<td>limit</td>
<td>limits the display to the first n entries where n is a value from 1 to 9999. The limit value can only be specified as the third parameter; for example: <strong>CSMON SUM ASID 10</strong> The default is all entries.</td>
</tr>
<tr>
<td>STATUS</td>
<td>displays the status of the IBM Virtual Storage Management (VSM) data.</td>
</tr>
<tr>
<td>MAP</td>
<td>lists the common storage blocks attributed to a specific address space. The following describes the parameters that you can use with MAP:</td>
</tr>
<tr>
<td>asi</td>
<td>identifies the subject address space instance. The asi value is required when the MAP parameter is specified; see Identifying address spaces on page 47 for more information.</td>
</tr>
</tbody>
</table>

---

**Example 1**

To list the common storage blocks allocated by JES2, enter:

**CSMON,MAP JES2**
Example 2

To list the common storage blocks allocated by ASID 29, enter:

**CSMONT,MAP (29)**

<table>
<thead>
<tr>
<th>AMTCSI</th>
<th>AMTCSII (29)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTCSI</td>
<td>START - END</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00B4B198-00B4B21F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00BA8300-00BAAF6F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00BB74C0-00BB74FF</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00BB7500-00BB7FB7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>00F51020-00F5105F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>01E3EF50-01E3EFA7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>01E3EFA8-01E3EFFF</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>020F1000-020F1027</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0215B000-0215B057</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0215B058-0215B0AF</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222B038-0222B08F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222B090-0222B0E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222C038-0222C08F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222C090-0222C0E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222D038-0222D08F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222D090-0222D0E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222E038-0222E08F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222E090-0222E0E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222F038-0222F08F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>0222F090-0222F0E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>02230038-0223008F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>02230090-022300E7</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>02231038-0223108F</td>
</tr>
<tr>
<td>AMTCSI</td>
<td>02231090-022310E7</td>
</tr>
</tbody>
</table>

Example 3

To display the status of VSM data collection, enter:

**CSMONT STATUS**

**AMTCSI** CSMON is using IBM VSM data. CSA=ON, SQA=ON AMT001A SYSPROG

Example 4

To display the ten address spaces using the most SQA, enter:

**CSMONT**

84  MainView SYSPROG Services User Guide and Reference
The CTCB (CTC) service provides the ability to abnormally terminate any task in any address space.

Optionally, you can specify

- the ABEND and reason code
- that a dump be taken
- that the task be allowed to retry

You can also indicate that the job step is to be abnormally terminated if the task does not recover.

You can use the TCB service to display the tasks in an address space and create a symbolic name for each TCB.

CTCB,@TCB4,S0C4,RETRY

**Syntax**

```
CTCB  icbadrs  asi
    Unnnn  Unnnn  Unnnn
    Sxxx   Sxxx   Sxxx
    RSNnnn RSNnnn RSNnnn
    DUMP   DUMP   DUMP
    RETRY  RETRY  RETRY
    STEP   STEP   STEP
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tcbadrs</strong></td>
<td>is a symbolic name that resolves to a TCB address in some address space (see examples) or the hexadecimal address of a TCB. If you specify a hexadecimal address, you must also specify the address space name as the second operand.</td>
</tr>
<tr>
<td><strong>asi</strong></td>
<td>defines the target address space. asi is an abbreviation for address space instance; see “Identifying address spaces” on page 47 for more information. If the first parameter is a symbol and its address space attribute defines the desired target address space, you do not need to provide the asi parameter. If you do provide a value for the asi parameter, the value overrides the address space attribute of the symbol. If the first parameter is a TCB address, you must specify a value for asi (the second parameter).</td>
</tr>
<tr>
<td><strong>Unnnn</strong></td>
<td><em>(optional)</em> can be entered in any position except the first. The letters nnnn must be replaced with a four-digit decimal user ABEND code with a value of 1-4095. The task will be abnormally terminated with this code. The default code is U0086.</td>
</tr>
<tr>
<td><strong>Sxxx</strong></td>
<td><em>(optional)</em> can be entered in any parameter position except the first. The letters xxx must be replaced with a three hexadecimal character system code. The task will be abnormally terminated with this ABEND code. If you specify both Unnnn and Sxxx, the one entered last will take precedence.</td>
</tr>
<tr>
<td><strong>RSNnnnn</strong></td>
<td><em>(optional)</em> can be entered in any parameter position except the first. The letters nnnn must be replaced by one to nine decimal digits. The decimal value is used as the reason code.</td>
</tr>
<tr>
<td><strong>DUMP</strong></td>
<td><em>(optional)</em> indicates that a dump is requested. <strong>Note:</strong> A dump will not be produced unless SYSUDUMP, SYSABEND, or SYSMDUMP has been preallocated. A dump will not be produced if suppressed by a recovery routine established by the task being terminated. Your installation can also suppress dumps.</td>
</tr>
<tr>
<td><strong>RETRY</strong></td>
<td><em>(optional)</em> indicates that the task's recovery routines should be allowed to retry; retry will not be allowed if RETRY is not specified.</td>
</tr>
<tr>
<td><strong>STEP</strong></td>
<td><em>(optional)</em> indicates that the jobstep should also be terminated if the task fails to recover.</td>
</tr>
</tbody>
</table>

**Example 1**

After using the TCB service to display information for the TCBs in an address space, you decide that you want to abnormally terminate the TCB labeled @TCB4. You do not care what abnormal termination code is used, so you accept the default of U0086. You also do not want the task’s recovery routine to be able to retry, you do not want
a dump, and you do not want to terminate the jobstep task; you only need to specify the symbolic name for the TCB.

CTCB,@TCB4

**Example 2**

You want to cause an abnormal termination in a task to test the recovery routine. Therefore, you want to specify a specific ABEND code, and you want to allow the task to retry:

CTCB,@TCB004,S0C4,RETRY

**Example 3**

You want to abnormally terminate a task with a user ABEND code, and you also want to provide a reason code:

CTCB,@TCB4,U2000,RSN30

**Example 4**

You want to abnormally terminate a task. You know the TCB address, and you know that the ASID for the address space is decimal 31. You can also enter the ASID as a hexadecimal value (X1F):

CTCB,160FF8,(31)

### DEVIATN

The DEVIATN (DEV) service checks the list of job names and the expected operational time range that you specified.

The DEV service also displays a message for each job that is not executing during the specified time range.

The DEV service obtains the list of jobs from one of the following places:

- the logical PARMLIB concatenation
- an existing set of data sets, which is represented by a DDNAME
- from a data set that you specified

The DEV service retains the location for the job list and uses the location on subsequent executions of the service, unless you specify a new value. The member name must begin with CONFIGxx followed by a user-specified one or two-
alphanumeric suffix. The default suffix is 00. For more information, see “DVTN” on page 314.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>default suffix of the CONFIGxx member</td>
</tr>
<tr>
<td>xx</td>
<td>suffix of the CONFIGxx member in SYS1.PARMLIB that is to be used</td>
</tr>
<tr>
<td>*</td>
<td>initial default value of the second parameter</td>
</tr>
<tr>
<td></td>
<td>The asterisk (*) indicates that the DEV service will obtain the member from the logical PARMLIB concatenation.</td>
</tr>
<tr>
<td>ddname</td>
<td>DDNAME of a pre-allocated partitioned data set, or group of concatenated partitioned data sets, that contains the member</td>
</tr>
<tr>
<td></td>
<td>This value must be eight characters or less.</td>
</tr>
<tr>
<td>dsname</td>
<td>name of a partitioned data set that contains the member</td>
</tr>
<tr>
<td></td>
<td>The data set is dynamically allocated and freed each time you execute the service. The data set name must contain a period (.).</td>
</tr>
</tbody>
</table>

**Note**

The DEV service retains the value that you specify for the second parameter and uses it as the default value on later executions of the DEV service until you specify a new value.

**SYNTAX for the CONFIGxx member**

The entries in the CONFIGxx member are entered as comment statements so that they can be included in the IBM PARMLIB member CONFIGxx.

The statements can also reside in other data sets. Each entry must use the following format:

*/NAME=identifier TIME=hhmm-hhmm [ACT=command]
The statement must begin in column one of the data set. Separate the keyword parameters and their values with spaces. The identifier and time range values are required and the DEV service ignores the third parameter, ACT=command, if it is specified.

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| identifier | job name (or Started Task ID for Started Tasks) of the job to be monitored. You can specify one of the following values:  
  - jobname  
  - jobname.stid  
  - .stid |
| hhmm-hhmm | beginning and ending values for the interval when the specified job must be active where:  
  - hh represents the hour (represented as a twenty four-hour clock value)  
  - mm represents the minute within the hour |
| ACT=command | an optional parameter that when present, is ignored by the DEV service however, this value is supported by the Exception Manager DVTN sampler |

For example, if the jobname is ABC and the started task ID is DEF then you could specify ABC, ABC.DEF or .DEF.

If the job name and the started task ID are not unique names (in other words they are also used by other address spaces), you must specify both the job name and started task ID.

The first hhmm value is the beginning of the range and the second hhmm entry represents the end of the range. Both values are required. The beginning value must be a number that is less than the ending value number. A beginning value of zero indicates 24-hour operation.

**Example**

This example is based on two assumptions:

- The started task is named ABC.DEF needs to be active from 10:00 AM to 10:00 PM and the job named DAYEND should execute between 11:00 PM and 11:10 PM.

- The name of the member containing the following statements is named CONFIGDP and the member is in a data set in the concatenation defined by DDNAME BBPARM.
The control statements for this situation are:

- /*NAME=ABC.DEF TIME=1000-2200
- /*NAME=DAYEND TIME=2300-2310

The command to execute the DEVTN service is:

**DEVTN,DP,BBPARM**

If you enter this command when either job should be executing and it is not, the DEVTN service issues a message similar to the following:

```
AMTCF81 WARNING ABC.DEF should be active 1000-2200
```

## DFLIST

The DFLIST service provides the following functions:

- When no operand is specified, DFLIST displays a list of all of the installed IBM facilities disclosed by the Store Facility List Extended instruction.
- When one or more facility numbers are specified as operands, DFLIST displays the name of each facility and its status (installed or not installed).

### Syntax

```
DFLIST nnn  [nnn]  [...,nnn]
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnn</td>
<td>facility number that is listed under the heading &quot;Facility Indications&quot; in chapter 4 (Control) of IBM Publication <em>z/Architecture Principles of Operation</em></td>
</tr>
</tbody>
</table>

### Example 1

To display the description and status of feature number 44, type

**DFL 44**

```
AMTDF1I  Num  Description
AMTDF2I  -------  "Perform Floating-Point Operation (PFPO) facility"
AMTDF4I  44  Perform Floating-Point Operation (PFPO) facility
AMTDF5I  is installed
AMT001A  SYSPROG
```
### Example 2

To display the description and status of features 40, 67, and 68, type

```
DFL 40,67,68
```

<table>
<thead>
<tr>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Set-program-parameter facility is installed</td>
</tr>
<tr>
<td>67</td>
<td>CPU-measurement counter facility is NOT installed</td>
</tr>
<tr>
<td>68</td>
<td>CPU-measurement sampling facility is NOT installed</td>
</tr>
</tbody>
</table>

### Example 3

To display a list of the installed features, type

```
DFL
```

<table>
<thead>
<tr>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;N3&quot; instructions feature</td>
</tr>
<tr>
<td>1</td>
<td>z/Architecture mode installed</td>
</tr>
<tr>
<td>2</td>
<td>z/Architecture mode is active</td>
</tr>
<tr>
<td>3</td>
<td>DAT Enhancement Facility</td>
</tr>
<tr>
<td>4</td>
<td>Invlalidate DAT Table Entry (IDTE) select 1</td>
</tr>
<tr>
<td>6</td>
<td>ASN Reuse Facility</td>
</tr>
<tr>
<td>7</td>
<td>Store Facility List Extended Feature</td>
</tr>
<tr>
<td>8</td>
<td>Enhanced DAF Facility</td>
</tr>
<tr>
<td>9</td>
<td>Sense-Running Status Facility</td>
</tr>
<tr>
<td>10</td>
<td>Conditional SSKE Feature</td>
</tr>
<tr>
<td>11</td>
<td>Configuration-Topology Feature</td>
</tr>
<tr>
<td>16</td>
<td>Extended-Translation Facility 2</td>
</tr>
<tr>
<td>17</td>
<td>Message Security Assist Feature</td>
</tr>
<tr>
<td>18</td>
<td>Long-displacement Facility high performance</td>
</tr>
<tr>
<td>20</td>
<td>The HFP-Multiply and Subtract facility</td>
</tr>
<tr>
<td>21</td>
<td>Extended-Immediate facility</td>
</tr>
<tr>
<td>22</td>
<td>Extended-translation facility 3</td>
</tr>
<tr>
<td>23</td>
<td>HFP-unnormalized-extension facility</td>
</tr>
<tr>
<td>24</td>
<td>ETF2-enhancement facility</td>
</tr>
<tr>
<td>25</td>
<td>Store-clock-fast facility</td>
</tr>
<tr>
<td>26</td>
<td>Parsing-enhancement facility</td>
</tr>
<tr>
<td>27</td>
<td>Move-with-optional-specifications facility</td>
</tr>
<tr>
<td>28</td>
<td>TOD-clock-steering facility</td>
</tr>
<tr>
<td>30</td>
<td>Extract-CPU time facility</td>
</tr>
<tr>
<td>31</td>
<td>Compare-and-swap-and store facility</td>
</tr>
<tr>
<td>32</td>
<td>Compare-and-swap-and store facility 2</td>
</tr>
<tr>
<td>34</td>
<td>General-instructions-extension facility</td>
</tr>
<tr>
<td>35</td>
<td>Execute-extensions facility</td>
</tr>
<tr>
<td>41</td>
<td>Floating-point-support-enhancement facilities</td>
</tr>
<tr>
<td>42</td>
<td>Decimal-floating-point facility (DFP)</td>
</tr>
<tr>
<td>43</td>
<td>DFP high performance</td>
</tr>
<tr>
<td>44</td>
<td>Perform Floating-Point Operation (PFPO) facility</td>
</tr>
</tbody>
</table>

### DONTSWAP

The DONTSWAP (DON) service makes an address space nonswappable.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>specifies your own address space</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

*Note*
If you do not specify an address space to be made nonswappable, your address space is changed.

**Example**

To make address space INVENTORY nonswappable, type

```
DONTSWAP INVENTORY
```

1 JES job ID, name, and current step name of the specified address space

2 service class

3 nonswappability count; the operating system keeps this count for the address space
   (If the count is greater than zero, the address space is still nonswappable; when the count reaches zero, the address space is swappable.)

**DSNAME**

The DSNAME (DSN) service displays information about a specified data set.

This information includes:
- volume serial number
- device type
- tracks allocated
- tracks used
- number of extents
- data set organization
- record format
- block size
- logical record length

**Syntax**

```
DSNAME dname
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dname</code></td>
<td>data set name</td>
</tr>
</tbody>
</table>

**Examples**

To display information about data set SYS1.DUMP05, type

```
DSNAME SYS1.DUMP05
AMTD10I DATASET SYS1.DUMP05 [1]
AMTD13I VOL=DUMP01; DEVT=3380; TRACKS ALLOCATED= 2175, USED= 920 [2]
AMTD14I EXTENTS= 1; [3]
AMTD14I DSORG=PS; RECFM=FB; BLKSIZE= 4160; LRECL= 4160 [4]
```

To display the volume serial numbers of data set MASTER.TAPE, type

```
DSNAME MASTER.TAPE
AMTD10I DATASET MASTER.TAPE
AMTD11I VOLUMES 874008 316002 000381 [5]
```

**Legend:**

1. name of the specified data set
2. volume and device type on which the data set resides, as well as the space allocated and used
3 number of extents used

4 RECFM/LRECL/BLKSIZE and organization of the data set

5 only volume information displayed for data sets residing on tape or for a disk data set residing on a volume that is not currently mounted

Usage notes

■ Only the volume serial number and data set organization are provided for VSAM data sets.

■ For data sets on multiple mounted volumes, the usage data is displayed for each volume.

■ A maximum of 20 volume serial numbers can be displayed per command.

DSSUM

The DSSUM service displays the maximum number of permitted scope=common data spaces (MAXCAD), the current number of scope=common data spaces, the percentage of the limit in use, the maximum number of data spaces that have been in use since the system was IPLed, and the number of lost CAD ASTES.

It also displays the following information for each address space that owns any data spaces:

■ Jobname

■ ASID

■ number of scope= ALL, COMMON, and SINGLE data spaces that are owned by the address space

■ the total number of data spaces owned by the address space.

This information can be sorted by any column by specifying the column name as the first parameter.
Syntax

![Syntax Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON</td>
<td>sorts the information by the number of scope=COMMON data spaces, in decreasing order</td>
</tr>
<tr>
<td><em>hdrname</em></td>
<td>name of a column</td>
</tr>
</tbody>
</table>

**Example 1**

**DSSUM**

```
AMTS42I Percent Max Common Area Data Spaces in use= 52.00%
AMTS42I Maximum Common Area Data Spaces= 100
AMTS42I Common Area Data Spaces (CADS) high-water mark= 53
AMTS42I Common Area Data Spaces (CADS) in use= 52
AMTS42I Number of lost CADS ASTES= 0
AMTS2I
AMTS52I Table sorted by COMMON
AMTS2I
AMTS0I                          |<------- Scope ------>|
AMTS1I Jobname   ASID  JITOKEN      All   Common   Single    Total
AMTS2I --------  ----  -------- -------- -------- --------  -------
AMTS3I *MASTER*  0001  :001#BT      39        8        0       47
AMTS3I EYUX410   00B4  :04O2#X6       4        6        0       10
AMTS3I DC$RTCS   0064  :02M2#OO       0        4        0        4
AMTS3I OMVS      000F  :00F2#BW       5        3        12       20
AMTS3I CONSOLE   000A  :00A2#BT      63        3       11       76
AMTS3I XCFAS     0006  :062#BW       8        2       10       20
AMTS3I JES2AUX   0041  :01Q2#C6       9        2        0       11
AMTS3I MAB5195   00BB  :04V2#X$       0        1        0        1
AMTS3I AAOHD11   0172  :09J3U2I     39        8        0       47
AMTS3I LDTTPAY5  0050  :0223ZF2     39        8        0       47
AMTS3I DEVMAN    000E  :00E2#BW      39        8        0       47
AMTS3I DC$TCPID  009E  :0422#TY      39        8        0       47
AMTS3I SMCO      0025  :00E2#NL      39        8        0       47
AMTS3I AAGNS72   015C  :08G3#B0      39        8        0       47
AMTS3I DC$NFSC9  0033  :0122#N@      39        8        0       47
AMTS3I ANTS0000  000D  :00D2#C6     39        8        0       47
AMTS3I SVOSP     0076  :0312#RU     39        8        0       47
```

**Example 2**

**DSSUM ASID**

```
AMTS42I Percent Max Common Area Data Spaces in use= 44.00%
AMTS42I Maximum Common Area Data Spaces= 100
AMTS42I Common Area Data Spaces (CADS) high-water mark= 45
AMTS42I Common Area Data Spaces (CADS) in use= 44
AMTS42I Number of lost CADS ASTES= 0
AMTS2I
AMTS52I Table sorted by ASID
AMTS2I
AMTS0I                          |<------- Scope ------>|
```

Chapter 4 Services  95
### DUMP

The DUMP (DUM) service displays the contents of storage in hexadecimal and character formats.

#### Syntax

![Diagram of DUMP syntax](image)

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>see “Expressions” on page 48</td>
</tr>
<tr>
<td>length</td>
<td><em>(optional)</em> length, in bytes, of the storage area to be dumped. The length can be an expression that evaluates to a decimal value in the range of 1-16,384. If you do not specify a length value, the value implied by the address expression will be used. If no length is implied by the address expression, the default is 40 hexadecimal (64 decimal). See “Expressions” on page 48 for more information. All values are assumed to be hexadecimal unless you append N to the value to indicate a decimal. For example, 64N equals 40.</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

When omitted, the address space attribute of the expression is used as the target address space. If no address space attribute is associated with the expression, your address space will be used as the target address space.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>displays the predefined and user-defined symbols, along with their respective definitions, that can be used with DUMP (see “EQUATE” on page 109 for information about defining symbols)</td>
</tr>
</tbody>
</table>

**Note**

The length and address space instance are shown as the second and third parameters in the syntax diagram. However, you can specify them in either order.

### Examples

To display the beginning of the CVT, type either one of the following:

#### Example 1

**DUM,10?**

```
AMTC1ZI 00000000_00FCC610  KEY= 0    BMVDWP4  (ASID 0125)
```

#### Example 2

**DUM,CVT**

```
AMTC1ZI 00000000_00FCC610  KEY= 0    BMVDWP4  (ASID 0125)
```

**Legend:**

1. display address
2. virtual storage key (first digit); a P following this digit indicates that the storage is fetch-protected; the remaining three bits are always zero
3. character representation of hexadecimal dump
Example 3

When you type DUMP without any operands, you see a list of all of the user-created symbols (by way of the EQUATE service) or the symbols created on behalf of the user (by the MEMSCAN, OLIST, and TCB services). For example, type

DUMP

AMTC1HI USER DEFINED SYMBOLS ARE:
AMTC1GI MOBJ0001 00000200_00000000
AMTC1GI MOBJ0002 00000001_00000000
AMTC1GI MOBJ0003 00000001_08200000
AMTC1GI @TCB001 7FD968
AMTC1GI @TCB001 7CA090
AMTC1GI @TCB003 7CA930
AMTC1FI @TCB004 7BCE88
AMTC1GI @TCB005 7CA5F0
AMTC1GI @TCB006 7EE680
AMTC1GI @TCB007 7EEA40
AMTC1GI @TCB008 7EE468
AMTC1GI @TCB009 7EE150

To display the symbols that can be used with the DUMP service, type

DUMP LIST

AMTC1HI USER DEFINED DUMP SERVICE LABELS ARE:
AMTC1GI @PWVT 00008C88
AMTC1GI @AVT 00009000
AMTC1GI @BBCT @AVT+58?
AMTC1FI PREDEFINED DUMP SERVICE Control BLOCK LABELS ARE:
AMTC1GI @ACB 10%+100?+14%+18?
AMTC1GI @AMCBS 10%+100?
AMTC1GI @ASCB 224%
AMTC1GI @ASMVT 10%+2C0?
AMTC1GI @JSTCB 224%+6C%+8%+7C%

Example 4

You can use a combination of predefined symbols and indirection indicators to display the CDEs for the job pack area (JPA) for an address space. For example, to locate the beginning of the chain and display the first CDE for address space JES2, type

DUMP,JSTCB+2C?,20,JES2

AMTC1ZI 00000000_005C1850 KEY= 0 JES2 (ASID 001A)
AMTC1ZI 005C1850 005D1120 00000000 C9C5C6C3 D5C1D4E2 *......IEFCNAMS*
AMTC1ZI 005C1860 8022A9E8 005D0868 000210FC 31224000 *.zY.)..........*

The +2c? in the command displays the 31-bit address located at +2C from the beginning of the area.
To display the next CDE in the chain, type

DUMP,*?,20*

AMTC1ZI 00000000_005D1120 KEY= 0 JES2 (ASID 001A)
AMTC1ZI 005D1120 005FF0000 00000000 C9C5C6E5 CBF14040 *.0.....IEFVH1 *
AMTC1ZI 005D1130 970E2F4B 0056C2A8 000210FC 31224000 *p......By.......*
To display the next CDE in the chain, simply repeat the previous command by entering an asterisk; for example, type

```
AMTC1ZI 00000000_005FF008  KEY= 0    JES2     (ASID 001A)
AMTC1ZI 005FF008 005FF150 00000000  *..1&....*
AMTC1ZI 005FF010 C9C5C1E5 D4F7FF0F2 97030900 005FF0F0  *IEAVM702p.....00*
AMTC1ZI 005FF020 000110FC 31224000  *...... .        *
```

You can repeat this command until all CDEs in the chain have been displayed.

**Example 5**

To evaluate an expression and display the storage at the resulting address, type

```
DU,MOBJ0009+740!,100
```

```
AMTC1XI MOBJ0009+90F0                Size= 4095M, Guard= 1M
AMTC1ZI 00000001_001090F8  KEY= 0    DHO2DBM1 (ASID 009E)
AMTC12I 001090F8                     00F20030 00000000          *.2......*
AMTC12I 00109100  00000000 01000000  00000001 22903000  *................*;
AMTC12I 00109110  00000000 00000000  00000000 143B8FE4  *...............U*
AMTC12I 00109120  BCE2BF0E AF590A68  C6C2D3D2 0F87E570  *.S......FBLK.gV.*
AMTC12I 00109130  00000001 00109000  00000000 00000000  *................*
AMTC12I 00109140  00000000 00000000  00000000 00000000  *..............+..*
AMTC12I 00109150  00000000 00000000  00000001 001091D0  *..............j.*
AMTC12I 00109160  C6D9C5C5 00000000  00000000 00000000  *FREE............*
AMTC12I 00109170  00000000 00000000  00000000 00000000  *................*
AMTC12I 00109180  00000000 00000000  00000000 00000000  *........FBLK....*
AMTC12I 00109190  C6D9C5C5 00000000  00000000 00000000  *FREE............*
AMTC12I 001091A0  00000000 00000000  00000000 00000000  *FREE............*
AMTC12I 001091B0  00000000 00000000  00000000 00000000  *FREE............*
AMTC12I 001091C0  00000000 00000000  00000000 00000000  *FREE............*
AMTC12I 001091D0  00000000 00000000  00000000 00000000  *FREE............*
AMTC12I 001091E0  00000000 00000000  00000000 00000000  *FREE............*
AMTC12I 001091F0  00000000 00000000  00000000 00000000  *FREE............*
```

**Note**

The resulting storage is within a memory object. See “OLIST” on page 176 for additional information about working with memory objects.

**Usage notes**

- The DUMP service operates in SRB mode when displaying data. Therefore, PSATOLD (task control block -TCB old pointer of the prefixed saved area) always contains zero and cannot be used to locate the current TCB (task control block).

- ZAP is sometimes used immediately following DUMP. The beginning address of the storage area last displayed by DUMP is retained. The ZAP service can refer to this address by using an asterisk (*) for the hexadecimal location parameter.

- For processor complexes consisting of more than one CPU, you can specify a CPU number. Add /c to the end of the value specified for *expression*, where c is the number of an available CPU.

- LIST or LI is a valid alias for the DUMP service.
If you omit the length parameter on the COMMAND line, you can specify the asi parameter in its place; for example, DUMP @CSCB INVENTORY, where INVENTORY is an address space instance.

**EDTINFO**

The EDTINFO (EDT) service provides information about unit names in the current Eligible Device Table (EDT).

Unit names are subdivided into generic and esoteric:

- Generic names are IBM device type names such as 3380, 3390, or 3400-6.
- Esoteric names are user defined (for example, SYSDA, TAPE, and CART).

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>classname</code></td>
<td>displays a list of the generic and esoteric unit names for the specified class name. You must prefix the class name with an equal (=) sign. The default is =DASD. Supported class names are DASD, TAPE, UREC, COMM, CTC, CHAR, and DISP.</td>
</tr>
<tr>
<td><code>unitname</code></td>
<td>displays the device number, device type, volser, and the use status of the specified unit name. The rest of these parameters are valid only when the first parameter is <code>unitname</code>.</td>
</tr>
<tr>
<td>ALL</td>
<td>displays information for all devices</td>
</tr>
<tr>
<td>Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PUBLIC</td>
<td>for DASD devices, limits the display to devices that contain volumes with a use attribute of PUBLIC; this parameter is ignored for other device types</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>for DASD devices, limits the display to devices that contain volumes with a use attribute of PRIVATE; this parameter is ignored for other device types</td>
</tr>
<tr>
<td>STORAGE</td>
<td>for DASD devices, limits the display to devices that contain volumes with a use attribute of STORAGE; this parameter is ignored for other device types</td>
</tr>
<tr>
<td>ONLINE</td>
<td>limits the display to online devices (default)</td>
</tr>
<tr>
<td>OFFLINE</td>
<td>limits the display to offline devices</td>
</tr>
</tbody>
</table>

### Example 1

To display the generic and esoteric unit names for device class DASD, type `EDTINFO`

<table>
<thead>
<tr>
<th>AMTED1I</th>
<th>DEVICE</th>
<th>[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTED2I</td>
<td>UNITNAME</td>
<td>COUNT</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>2305-2</td>
<td>1</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>3390</td>
<td>144</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>ALLDA</td>
<td>608</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>SYSTSO</td>
<td>384</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>SYSVIO</td>
<td>1</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>TSGDA</td>
<td>16</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>VIO</td>
<td>1</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>VI03390</td>
<td>1</td>
</tr>
<tr>
<td>AMTED3I</td>
<td>SYSALLDA</td>
<td>660</td>
</tr>
</tbody>
</table>

**Legend:**

1. device class being displayed
2. number of devices defined for the respective generic or esoteric unit name
3. whether the unit name is VIO eligible

### Example 2

To display information about the online devices for the generic unit name BABDA, type `EDTINFO, BABDA`

<table>
<thead>
<tr>
<th>AMTED4I</th>
<th>10 Devices for Generic unitname BABDA</th>
<th>[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTED7I</td>
<td>Displaying only: online private public storage</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTED9I</td>
<td>DVN</td>
<td>Dev Type</td>
</tr>
<tr>
<td>AMTED5I</td>
<td>8519</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851A</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851B</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851C</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851D</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851E</td>
<td>3390</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851F</td>
<td>3390</td>
</tr>
</tbody>
</table>
Legend:

1 number of devices, whether the unit name is generic or esoteric, and the unit name

2 how the data was filtered

3 device number

4 device type

5 volume serial number

6 the IBM "Internal Lookup Value" (LUV)

7 Mount status: Private, public, or storage. If both online and offline devices are being displayed, also indicates if the device is online or offline

Example 3

To display information for all of the devices (online and offline) with the unit name HDSBASE, type

EDITINFO, HDSBASE, BOTH
AMTED6I 630C 3390 3010200F offline
AMTED6I 630D 3390 3010200F offline
AMTED6I 630E 3390 3010200F offline
AMTED6I 630F 3390 3010200F offline
AMTEDBI 32 devices were selected
AMT001A SYSPROG

Legend:

1 the number of devices, whether the unit name is generic or esoteric, and the unit name

2 message AMTED7I was not displayed because data is not filtered

3 device number

4 device type

5 volume serial number

6 IBM’s "Internal Lookup Value" (LUV)

7 Mount status: Private, public, or storage. IF both online and offline devices are being displayed, also indicates if the device is online or offline

Example 4

To display all DASD volumes with a volume serial number that begins with BAB35 type

EDITINFO, SYSALLDA IVOL (BAB35*)

<table>
<thead>
<tr>
<th>DVN</th>
<th>Dev Type</th>
<th>VOLSER</th>
<th>LUV</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>8515</td>
<td>3390</td>
<td>BAB351</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8521</td>
<td>3390</td>
<td>BAB350</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8523</td>
<td>3390</td>
<td>BAB352</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8529</td>
<td>3390</td>
<td>BAB353</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>852A</td>
<td>3390</td>
<td>BAB354</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>852B</td>
<td>3390</td>
<td>BAB355</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>852E</td>
<td>3390</td>
<td>BAB358</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8555</td>
<td>3390</td>
<td>BAB359</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8596</td>
<td>3390</td>
<td>BAB356</td>
<td>3010200F</td>
<td>Private</td>
</tr>
<tr>
<td>8597</td>
<td>3390</td>
<td>BAB357</td>
<td>3010200F</td>
<td>Private</td>
</tr>
</tbody>
</table>

AMTEDBI 10 devices were selected
AMT001A SYSPROG

Example 5

To display only those devices with a generic unit name of 3390 that are in storage status, type

EDITINFO, 3390, STORAGE

AMTED4I 17,320 Devices for Generic unitname 3390
AMTED7I Displaying only: online storage
AMTED0I
Usage notes

- You can abbreviate the class name to the shortest unique value, as illustrated in the following examples:
  - =T or =TA (for TAPE)
  - =U or =UR (for UREC)

- Unlike class names, unit names cannot be abbreviated.

- You can include devices in multiple unit names.

- For DASD devices, one of these use statuses is displayed:
  - PRIVATE
  - PUBLIC
  - STORAGE


ENCLAVE

The ENCLAVE (ENC) service locates active enclaves.

For each located enclave, this service displays name and ASID of the owning address space

- whether the enclave is dependent or independent

- whether the enclave is local or foreign

- the enclave's service class

This service also provides the ability to QUIESCE, RESUME, and change the service class of local-independent.
Syntax

To change the service class of an enclave, specify both the first and second positional parameters. The first positional parameter (parm1) may be either a 16 character enclave token or a symbolic name assigned by the ENCLAVE service when it was used to display the active enclaves. Note: Symbolic names are only assigned to local-independent enclaves because they are the only ones that can be modified.

Example 1

To display the active enclaves and their assigned symbolic names (label), type

```
ENCLAVE
```

<table>
<thead>
<tr>
<th>AMTEN11</th>
<th>Owner Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTEN21</td>
<td>Enclave Token Stat ASID Job Name Sys Name Class Label</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003C00000008 1L 0094 DC$TCPD2 SJSD SYSTOTHER ENC#1</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003400000006 1L 008D DC$TCPID SJSD SYSTOTHER ENC#2</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000006400000016D 1L 010C WS70BSS SJSD WEBMED ENC#3</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002400000002 DL 0017 AXR SJSD STCNRM</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002800000003 DL 002D DC$PAS SJSD STCPAS</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003800000007 DL 0094 DC$TCPD2 SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000005 DL 008D DC$TCPID SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000006800000379 DL 0044 DMRRRS01 SJSD BATLOW</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002000000001 DL 002D DC$PAS SJSD STCPAS</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000004 DL 0034 MIMGR SJSD SYSTEM</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003800000007 DL 0094 DC$TCPD2 SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000005 DL 008D DC$TCPID SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000006800000379 DL 0044 DMRRRS01 SJSD BATLOW</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002000000001 DL 002D DC$PAS SJSD STCPAS</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000004 DL 0034 MIMGR SJSD SYSTEM</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003800000007 DL 0094 DC$TCPD2 SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000005 DL 008D DC$TCPID SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000006800000379 DL 0044 DMRRRS01 SJSD BATLOW</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002000000001 DL 002D DC$PAS SJSD STCPAS</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000004 DL 0034 MIMGR SJSD SYSTEM</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003800000007 DL 0094 DC$TCPD2 SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000005 DL 008D DC$TCPID SJSD SYSTSC</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000006800000379 DL 0044 DMRRRS01 SJSD BATLOW</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000002000000001 DL 002D DC$PAS SJSD STCPAS</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003000000004 DL 0034 MIMGR SJSD SYSTEM</td>
</tr>
<tr>
<td>AMTEN41</td>
<td>0000003800000007 DL 0094 DC$TCPD2 SJSD SYSTSC</td>
</tr>
</tbody>
</table>

Example 2

To change the service class of the enclave labeled "ENC#3" from WEBMED to WEBHI, type

```
ENCLAVE,ENC#3,WEBHI
```

```
AMTEN6I Enclave ENC#3 owned by WS70BSS (010C), ETOKEN=X'0000000640000016D'
AMTEN7I Service Class changed to WEBHI
AMTEN5I Tuesday, July 13, 2010; 5:06:48.67 PM Local
AMT001A SYSPROG
```

Example 3

To display the Quiesce enclave from the ENC label, type

```
ENCLAVE
```

```
AMTEN1A SYSPROG
```
ENQUEUES

The ENQUEUES (ENQ) service displays current enqueue conflicts or requested system enqueue information.

*Note*
The QUEUE service functionally replaces the ENQUEUES service, while providing new features. BMC recommends that you use the QUEUE service rather than the ENQUEUES service because BMC intends to remove the ENQUEUES service in a future release. The QUEUE service functionally replaces the ENQUEUES service, while providing new features.

An enqueue conflict exists when one or more jobs are waiting on a resource held by another job. This service helps you determine the cause of reduced system performance, the name and system ID for jobs holding a resource, and the jobs waiting for the same resource.

A conflict exists when one or more requesters are waiting for a resource. If this occurs frequently for a given resource, you can cause that resource to be omitted from the display by adding it to BBPARM member $$XENQ. See “$$XENQ” on page 32.

When you type the ENQUEUES command without operands, information about any current enqueue conflicts is displayed. The ENQUEUES service also displays enqueue information for a specified minor name and major name (R name and Q name, respectively), or all enqueue information for a specified major name (Q name). The default major name is SYSDSN. Therefore, you can display enqueues for a data set simply by typing the data set name (minor name).

**Syntax**

The parameters are defined as follows:
### Examples

The following examples demonstrate how the ENQUEUES service presents information about enqueue conflicts and resource ownership.

### Example 1 – Conflict information example

An enqueue conflict exists when one or more jobs wait for a resource held by another job. The ENQUEUES service displays the name and system ID of jobs holding a resource and the jobs waiting for the same resource. Use this information to determine the source of the performance delay in your system.

To display all current enqueue conflicts, type

**ENQUEUES**

```
AMTQ1PI SYSTEM (LOCAL) Q=SYSZVVDS R=CATALOG.ICFMCAT.SYSC
AMTQ1QI SYSID  JOBNAME   ASID   STAT TYP TIME 14:12:19
AMTQ1RI SYSB   QA7A     (005B)  OWNS EXC
AMTQ1RI SYSA   FLN1     (0096)  WAIT EXC
AMTQ1RI SYSB   QA7      (0352)  WAIT EXC
AMTQ1LI
AMTQ1PI SYSTEM (LOCAL) Q=SYSIGGV2 R=ICFUCAT.VTSG304
AMTQ1QI SYSID  JOBNAME   ASID   STAT TYP TIME 14:12:19
AMTQ1RI SYSB   ARG1     (00A2)  WAIT EXC
AMTQ1RI SYSC   MPP1     (0121)  OWNS SHR RES=003 PEND 291 TSG304 NR
```

**Legend:**

1. **scope of enqueue**
   - The possible scopes are as follows:
     - **SYSTEMS**
     - **SYSTEM**
     - **STEP**

2. **whether the resource is global or local**
3 major name (Q=)
4 minor name (R=)
5 system ID for system executing the task that is holding or waiting for the resource
6 job name, TSO user ID, or Started Task ID for the address space containing the task that is holding or waiting for the resource; the job name is not available in some situations
7 ID for the address space holding or waiting for the resource
8 current status (OWNS or WAIT), indicating that the task holds (OWNS) the resource or is waiting (WAIT) for the resource
9 type of enqueue: EXC for exclusive, SHR for shared
10 indicates that a reserve is associated with the enqueue
   The reserve count is also displayed unless it is zero, in which case the equal sign is also omitted. CVT is displayed instead of RES if the reserve has been converted to a global enqueue.
11 if present, indicates that the reserve is pending, which means that the task is waiting to reserve the devices; generally, the device is reserved by another system
12 device number in hexadecimal
13 volume serial number
14 if present, indicates that the device is not ready

Example 2 – Resource ownership information examples

Typing ENQUEUES followed by the minor name (R name) and major name (Q name) displays the jobs that hold the specified resource and information related to their use of the resource. A partial minor name displays all enqueues with minor names that begin with the specified characters for the requested major name. An asterisk typed as the last character indicates a partial minor name. SYSDSN is the default major name.

To display all enqueues for data sets starting with SYS1.L, type

```plaintext
ENQUEUES SYS1.L*
```

```
AMTQ1P1 SYSTEM (LOCAL) Q=SYSDSN R=SYS1.LPALIB
AMTQ1QI SYSID JOBNAME ASID STAT TYP TIME 16:05:20
AMTQ1RI SYSB MEE2 (00C0) OWNS SHR
```

```
AMTQ1P1 SYSTEM (LOCAL) Q=SYSDSN R=SYS1.LINKLIB
AMTQ1QI SYSID JOBNAME ASID STAT TYP TIME 14:12:19
AMTQ1RI SYSB LLA (0006) OWNS SHR
```
Besides SYSDSN, another common major name is SPFEDIT. ISPF EDIT uses the major name SPFEDIT and a 52-character minor name consisting of the data set name (a 44-character field padded with blanks), followed by the member name.

The ENQUEUES service provides special syntax to allow you to display these ISPF EDIT enqueues for specific members. Type the data set name (dsn), followed by a slash (/), followed by the member name, a comma (or blank), and the major name SPFEDIT.

For example, to determine the job names editing member IEASYS00 in SYS1.PARMLIB, type

```
ENQUEUES SYS1.PARMLIB/IEASYS00,SPFEDIT
```

<table>
<thead>
<tr>
<th>AMTQ1P1</th>
<th>SYSTEM (LOCAL)</th>
<th>Q=SPFEDIT R=SYS1.PARMLIB</th>
<th>IEASYS00</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQ1Q1</td>
<td>SYSID JOBNAME</td>
<td>ASID STAT TYP TIME 14:12:19</td>
<td></td>
</tr>
<tr>
<td>AMTQ1R1</td>
<td>SYSB CIR3</td>
<td>(0087) OWNS EXC</td>
<td></td>
</tr>
</tbody>
</table>

In each of these examples, the header line is followed by two or more information lines. The ENQUEUES service displays one line for each task that issued an ENQ or RESERVE for the resource.

To display all jobs that are using all resources for a major name, type an asterisk (or a comma to indicate no minor name), followed by the major name, as illustrated here:

```
ENQUEUES *,SYSZVVDS
```

<table>
<thead>
<tr>
<th>AMTQ1P1</th>
<th>SYSTEM (LOCAL)</th>
<th>Q=SYSZVVDS R=CATALOG.ICFMCAT.SYSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQ1Q1</td>
<td>SYSID JOBNAME</td>
<td>ASID STAT TYP TIME 14:12:19</td>
</tr>
<tr>
<td>AMTQ1R1</td>
<td>SYSB QA7A</td>
<td>(0058) OWNS EXC</td>
</tr>
<tr>
<td>AMTQ1R1</td>
<td>SYSB FLN1</td>
<td>(0096) WAIT EXC</td>
</tr>
<tr>
<td>AMTQ1R1</td>
<td>SYSB QA7</td>
<td>(0352) WAIT EXC</td>
</tr>
</tbody>
</table>

**Note**

Specifying a partial minor name can produce a large volume of output.

---

**EQUATE**

Several services permit the use of an expression as an operand to represent a storage location within an address space.

The EQUATE (EQU) service facilitates the use of expressions by permitting you to assign a symbolic name to represent an expression.

The EQUATE service also provides the ability to display, redefine, and delete symbol definitions. The symbol name must begin with an alphabetic or national character ($, #, or @) that can be followed by up to eight alphanumeric and national characters (the symbol name can be a maximum of nine characters).
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| symbol    | symbol to be defined or displayed  
The symbol must begin with @ and can be followed by one-to-eight alphanumeric characters. |
| LIST      | displays the definition of the symbol, if previously defined (default) |
| definition| expression that defines the symbol  
This expression can begin with either a previously defined symbol (including the one being defined) or a hexadecimal address, optionally followed by offset values (+ or -) or indirection indicators (%, ?, !, >) or both. For example, the start of the CVT prefix can be defined as 10%-100. |
| length    | (optional) defines the length attribute for the symbol  
The value is assumed to be in hexadecimal unless followed by the letter N. For example, to specify a value of 64, enter 40 or 64N. If this parameter is omitted and the definition begins with a previously defined symbolic name, the length attribute of that symbol is used. Otherwise, a value of 64N is used. |
| asi       | address space instance; see Identifying address spaces on page 47 for more information  
If this parameter is omitted, the address space attribute associated with the expression will be used if previously defined; otherwise, the job instance token for your address space will be used. |
| DELETE    | deletes the symbol |

Examples

To display the definition of symbol @JSCB, type

```
EQUATE,@JSCB
```

If this parameter is omitted, the address space attribute associated with the expression will be used if previously defined; otherwise, the job instance token for your address space will be used.

To define symbol @A as symbol @JSCB + DC? and give it a length of 64 bytes, type

```
EQUATE,@a,@JSCB+DC?,64N
```

If this parameter is omitted, the address space attribute associated with the expression will be used if previously defined; otherwise, the job instance token for your address space will be used.
To delete symbol @A, type

\texttt{EQUATE,@A,DELETE}

\texttt{AMTEQ4I @A DELETED}

\section*{ESCLASS}

The ESCLASS (ESC) service provides the ability to display a list of all current service classes.

It also provides the ability to

- display a specific service class
- change a service class
- quiesce an address space
- resume a job that has been quiesced

\subsection*{Syntax}

The parameters are defined as follows:

\begin{center}
\begin{tabular}{|l|l|}
\hline
\textbf{Parameter} & \textbf{Description} \\
\hline
\texttt{asi} & address space instance; see “Identifying address spaces” on page 47 for more information \\
\texttt{newclass} & new service class name to be applied to the address space \\
\texttt{QUIESCE (Q)} & address space is to be quiesced \\
\texttt{RESUME (R)} & service class for the specified address space is to be set according to the current service policy \\
\hline
\end{tabular}
\end{center}

\subsection*{Example 1}

To display a list of all service classes in the current service policy, type
Example 2

To display the service class and related service information for a specific address space, type

ESC,XTSTHPAS

Example 3

To change the service class for a specified address space and display the information, type
ESC,XTSTHPAS,STCNRM

AMTES3I Service class changed for address space XTSTHPAS
AMTES0I
AMTES1I Job name: XTSTHPAS
AMTES1I Service class: STCNRM
AMTES1I Description: Normal STC's
AMTES1I Current period: 1
AMTES2I Importance lvl: 4
AMTES2I Period type: Velocity goal
AMTES2I Goal: 30%

Example 4

To quiesce a specified address space, type

ESC,XTSTHPAS,Q

AMTES9I Address space XTSTHPAS has been quiesced.

Example 5

To resume (restart) a specified address space and display the information, type

ESC,XTSTHPAS,RESUME

AMTES8I Resume successful for address space XTSTHPAS
AMTES0I
AMTES1I Job name: XTSTHPAS
AMTES1I Service class: STCPAS
AMTES1I Description: PAS STC's
AMTES1I Resource group: PASSTC
AMTES1I Current period: 1
AMTES2I Importance lvl: 3
AMTES2I Period type: Velocity goal
AMTES2I Goal: 60%

Usage notes

The following points detail what the ESCLASS service can do.

- display a list of all service classes in the current service policy and their descriptions
  This display is produced when the ESCLASS service is executed without any operands.

- display the service class for a specific address space and service-related information
  This display is produced when a job name or ASID is specified as the first and only parameter. The ASID must be enclosed in parentheses and is assumed to be a decimal value unless preceded by an x.
  Example: (x24) = (36)

- change the service class for an address space
  This action is taken when a valid job name (or ASID) is specified as the first parameter and a valid service class is specified as the second parameter.
**Example**

ESCLASS,(x24), newclass will change the service class for the job in the address space with an ASID of decimal 36 (hexadecimal 24) to newclass.

- quiesce an address space by specifying a valid job name or ASID as the first parameter and Q or QUIESCE as the second parameter

  **Example**

  ESCLASS job1,Q will quiesce job1.

- resume (restart) a job that has been quiesced and cause the service class to be set according to the current service policy

  Specify the name of a job or an ASID as the first parameter and the keyword RESUME as the second parameter.

  **Note**

  Resuming a job causes the appropriate service class in the current service policy to be applied to the specified job. If you are resetting the job to reverse a QUIESCE, check the service class that is displayed to ensure that it is the desired service class. If not, use the ESCLASS service to change the service class. An example of resuming an address space:

  ESCLASS,JOB1,RESUME

---

**EXECUTE**

The EXECUTE (EXE) service executes a list of SYSPROG Services, MVS, or JES commands listed in a member within the specified data set.

This service enables you to invoke standard sequences of commands. Each SYSPROG Services command in the data set must be prefixed with @ (for example, @ENQ, @REPLIES) to distinguish it from MVS or JES commands.

**Syntax**

```
EXECUTE memberName dsname
```

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>memberName</code></td>
<td>member within the specified or default data set</td>
</tr>
<tr>
<td><code>dsname</code></td>
<td>specified data set name</td>
</tr>
</tbody>
</table>

- If you do not specify a data set name, EXECUTE will use the previously specified data set.
- If no previously specified data set exists, EXECUTE will use the data set allocated to DDNAME BBPARM.
- If no data set is allocated to DDNAME BBPARM, EXECUTE will use the data set allocated to DDNAME LIB.
- If no data set is allocated to DDNAME LIB, MainView for z/OS will display an error message.

**Example**

To execute the commands listed in BBPARM library member SHIFT1, type

```
EXE,SHIFT1
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTK11I</td>
<td>V 01A0,OFFLINE [1]</td>
</tr>
<tr>
<td>AMTQ2PI</td>
<td>SYSTEMS (GLOBAL) Q=SYSIGGV2 R=ICFMCAT.SYSB [2]</td>
</tr>
<tr>
<td>AMTQ2RI</td>
<td>SYSID JOBNAME ASID STAT TYP TIME 9:11:08</td>
</tr>
<tr>
<td>AMTK10I</td>
<td></td>
</tr>
<tr>
<td>AMTK11I</td>
<td>@RES [1]</td>
</tr>
<tr>
<td>AMTK10I</td>
<td></td>
</tr>
<tr>
<td>AMTK11I</td>
<td>@ENO [1]</td>
</tr>
<tr>
<td>AMTK10I</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. command read from the BBPARM member SHIFT1

2. results of the command

In the following example, the EXECUTE service is used to execute three services contained in member STATUS in the data set VAM3.RES.LIB. The services are preceded by the @ symbol, indicating that they are MainView SYSPROG services. The services are executed sequentially, with the results displayed after each service name.

To execute the contents of the member STATUS in the data set VAM3.RES.LIB, type

```
EXE,STATUS,VAM3.RES.LIB
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTK11I</td>
<td>@RES [1]</td>
</tr>
<tr>
<td>AMTQ2PI</td>
<td>SYSTEMS (GLOBAL) Q=SYSIGGV2 R=ICFMCAT.SYSB [2]</td>
</tr>
<tr>
<td>AMTQ2RI</td>
<td>SYSID JOBNAME ASID STAT TYP TIME 9:11:08</td>
</tr>
<tr>
<td>AMTK10I</td>
<td></td>
</tr>
<tr>
<td>AMTK11I</td>
<td>@ENO [1]</td>
</tr>
<tr>
<td>AMTK10I</td>
<td></td>
</tr>
<tr>
<td>AMTK11I</td>
<td>NO ENQ CONFLICTS EXIST [2]</td>
</tr>
</tbody>
</table>
Legend:

1 name of the service to be executed

2 results of the executed service

Usage notes

- The z/OS and JES2 commands are executed by using the COMMAND service. Each z/OS command that is executed is logged at the issuing console.

- The data set name specified (or the default data set) is not saved across sessions. You must specify a data set name, or accept the defaults, each time you restart MainView for z/OS.

- You can display the previously entered data set name by typing EXECute without any parameters.

EXIT

The EXIT (EXI) service terminates an address space even if it does not respond to CANCEL or STOP commands from the system.

Note

Use the EXIT service only as a last resort. The EXIT service uses CALLRTM to pass the job to the Recovery/Termination Manager for termination processing and does nothing to clean up after the target address space. As a result, CSA that was allocated by the address space might not be freed. Any recovery done is performed by the Recovery/Termination Manager.

Syntax

The parameter is defined as follows:
When MainView SYSPROG Services is operating in conversational mode, the EXIT service displays the following information to describe the selected job instance and issues message AMTE21A, requesting you to confirm that you want to terminate the described job instance. Reply U if you want to terminate the job instance. Any other reply will abort the request. Ensure that the described job instance is indeed the instance that you want to terminate prior to replying with U.

- Jobname
- ASID (in hexadecimal)
- Job Entry Subsystem Job ID
- When the job instance was initiated

**Example**

```exi
EXI,$WAIT
AMTE2AI Job instance $WAIT.W1 ASID(X0193) JES2 ID STC02868
AMTE2BI Initiated Thursday, September 23, 2010 at 7:28:05.10 PM Local
AMTE21A Reply U to terminate $WAIT.W1 ASID(X0193) JES2 ID STC02868
u
AMTE22I $WAIT.W1 ASID(X0193) JES2 ID STC02868 terminated by BMVDWP3
AMT001A SYSPROG
```

**FCOMMON**

The FCOMMON (FCO) service frees common storage (CSA or SQA) that should have been freed by a user or system application.

This service helps you regain common storage that was allocated by an application and not freed when the application finished.

**WARNING**

Use FCOMMON with extreme caution. Storage attributed to a specific address space instance might, in fact, be shared by multiple address spaces. Freeing storage that is still in use will cause the system to fail. Free storage only when you know that it is no longer in use.
Syntax

```
FCOMMON  address  length  spid
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>address of storage to FREEMAIN, in hexadecimal format</td>
</tr>
<tr>
<td>length</td>
<td>length of storage to FREEMAIN, in decimal format</td>
</tr>
<tr>
<td>spid</td>
<td>storage subpool identifier: 226, 227, 228, 231, 239, 241, or 245</td>
</tr>
<tr>
<td>key</td>
<td>key of storage: 0-15 for subpools 227, 228, 231, and 241</td>
</tr>
</tbody>
</table>

**Note**
The WHERE service can be used to determine the subpool and key of storage at a given address.

Example

Type

```
FCOMMON,0C84BC68,95128,228,4
```

AMTFCAI Reply Y/N to confirm request to FREEMAIN 0C84BC68 for 95,128 bytes

y

AMTFCAI FREEMAIN of requested area was successful

AMT001A SYSPROG

**FINDMBR**

The FINDMBR (FIN) service searches the specified data set concatenation or the LNKLST and displays the name of each data set that contains the member.

Syntax

```
FINDMBR  member  ddbname
```

The parameters are defined as follows:
### Examples

To list all members named MV in the SYSPROC concatenation for your address space, type

**FIND,MV,SYSPROC**

```
AMTFM1I JOBNAME: BMVDWP6  DDNAME: SYSPROC  MEMBER: MV
AMTFMVI ===--------------------------------------------------===
AMTFM2I CT   --Last Update-- USERID   Data Set Name
AMTFMLI   1  21Dec2004 18:21 BMVDWP1  VAM3.CLIST
AMTFMLI  13   9Mar2004 15:08 BTSSED1  SYS2.SHRD.CLIST
```

**Legend:**

1. job name
2. DD name
3. member name
4. concatenation number of library within the specified DD name
5. date that member was last updated
6. time that member was last updated
7. ID of user who updated the member last
8. name of data set

To locate a member in an address space’s LNKLST set, type

**FIND,BLDQS, LNKLST,BMVDWP4**

```
AMTFMCI Searching LNKLST Set  ipl
AMTFM1I Jobname: BMVDWP4  DDNAME: LNKLST  Member: BLDQS
AMTFMVI ===--------------------------------------------------===
AMTFM2I CT SIZE   ALIAS OF   AC       DATA SET
AMTFMLI  30 01908  ILBOQSU    00       CEE.SCEERUN
AMTFMLI  65 01840  ILBOQSU    00       SYS1.VSCOBII.SYS.COB2LIB
```

**Legend:**

- 1 job name
- 2 DD name
- 3 member name
- 4 concatenation number of library within the specified DD name
- 5 date that member was last updated
- 6 time that member was last updated
- 7 ID of user who updated the member last
- 8 name of data set
To list all RESXA proc names available through the alternate JES2, type

**FINDMBR, RESXA, PROCLIB, JES2A**

To list the data sets that are allocated with DD name LOADLIB to the address space DWPBPAS and contain load module LGS, type

**FINDMBR, LGS, LOADLIB, DWPBPAS**

Legend:

1. number of bytes in load module
2. alias name of member
3. authorization code for the module

**HELP**

The HELP (HEL) service displays information about the function and syntax of MainView SYSPROG Services.

**Syntax**

The parameter is defined as follows:
### Example 1

To display information about the JITOKEN service, type

```
HELP JT
```

**AMTH11I SERVICE**: JITOKEN

**AMTH11I FUNCTION**: Given a job name or ASID, JITOKEN returns a "Job Instance Token" that uniquely identifies the job instance. This token can subsequently be used in place of the job name in any service to assure that the intended action is only taken against that specific job instance.

**AMTH11I**

**AMTH11I**

**AMTH11I**

Job Instance Tokens always begin with a colon (:).

**AMTH11I**

**AMTH11I**

**AMTH11I**

**SYNTAX**: 

```
>>-JITOKEN------jobname---------------------------------------------><
```

```
|--(nnnn)---|
```

```
+-----------------------------------+--------------------------------+
```

```
| ENTER...                          | TO...                          |
```

```
AMTH11I | jit  jobname                      | Get a Job Instance Token for   |
```

```
AMTH11I | the job "jobname".                |
```

```
AMTH11I | jit  (476)                        | Get a Job Instance Token for   |
```

```
AMTH11I | the current job in address space  |
```

```
AMTH11I | 476 (decimal).                    |
```

```
AMTH11I | jit  (X476)                       | Get a Job Instance Token for   |
```

```
AMTH11I | current job in address space      |
```

```
AMTH11I | 476 (hexadecimal).                 |
```

**AMTH001A SYSPROG**

### Example 2

To display a list of all MainView SYSPROG Services, type

```
HELP
```

<table>
<thead>
<tr>
<th><strong>AMTH11I</strong> AEW</th>
<th>Alias of WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AMTH11I</strong> APF</td>
<td>Lists, adds, deletes, changes or restores entries in APF list</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> ASM</td>
<td>Lists information about auxiliary storage &amp; paging data sets</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> ASVT</td>
<td>Displays address space vector table information</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> AUTHTSO</td>
<td>Lists, builds, and restores TSO authorization lists</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> BBXS</td>
<td>Manages BMC Software subsystem services</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CDE</td>
<td>Lists the modules loaded in a specified address space</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CNSL</td>
<td>Simulates an MVS operator's console at a TSO user's terminal</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> COMMAND</td>
<td>Executes operator commands from a TSO terminal</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CONSOLES</td>
<td>Displays MCS console buffer usage</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CPU</td>
<td>Displays the proportionate use of CPU resources by job</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CSA</td>
<td>Displays the level of CSA and ECSA usage</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> CSMON</td>
<td>Displays the use of common storage by address space</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> DEVIATN</td>
<td>Displays deviation from a list of jobs in member CONFIGxx</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> DFLIST</td>
<td>Displays information about optional CPU features</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> DONTSWAP</td>
<td>Makes an address space nonswappable</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> DSNAME</td>
<td>Displays the volume serial number of a cataloged data set</td>
</tr>
<tr>
<td><strong>AMTH11I</strong> DSSUM</td>
<td>Displays Data Space information</td>
</tr>
</tbody>
</table>
IFA

The IFA service displays information about actual and potential IBM zSeries® Application Assist Processor (zAAP) usage for each address space that has work eligible for execution on zAAP processors.

Syntax

There are no parameters for the IFA service.

Example

Type

IFA

Legend:

1 name of address space

2 total CPU usage

3 actual IFA processor percentage of the total processor (CPU) usage

4 potential (eligible) IFA processor percentage of the total processor (CPU) usage

5 number of AMTIF6I messages (one per address space)
The INFO (INF) service displays information about the current operating environment.

This service lets you list the release level of the operating system, the serial number and model of each CPU, and the type of IPL last performed.

**Syntax**

Example 1

To display information about the current operating environment, type

```
INFO
```

```
AMTR0PI Program Event Recording (PER) SLIP trap DP01 is active
AMTR0HI System Name=SJSD
AMTR01I SYSPRG Release 3.8.00
AMTR02I z/OS 01.13.00 FMID JBB778H z/Architecture HiperDispatch=ON
AMTR03I CPU 0 Serial Number 04F037 Model 2827, Type=CP
AMTR03I CPU 1 Serial Number 04F037 Model 2827, Type=CP
AMTR03I CPU 2 Serial Number 04F037 Model 2827, Type=ZIIP
AMTR0FI LPAR name=SJSD, Max of config allowed= 12.5%
AMTR0GI Logical CPUs: Active= 2, Standby= 1, Reserved= 0
AMTR04I Last IPL was COLD Start (CLPA) on 7/17/2013 at 22:09:36
AMTR04I From SJSD00 on 8162
AMTR0BI IPL used LOADD0 in SYS1.IPLPARM on volume HCD002 (A440)
AMTR0CI IEASYM list = D1
AMTR0CI IEASYM list = (D1), Source=Oper
AMTR05I TSO Version 3 Release 13 Mod 0
AMTR06I VTAM Terminal ID TCPD0015
AMTR0EI Common Storage Tracking: CSA=ON  SQA=ON
AMTR07I Volser PARMLIB Data Set Name
AMTR08I --------------------------
AMTR09I SJSD00 RXA38.TEST.PARMLIB
AMTR09I SJSD00 SYS1.SJSD.PARMLIB
AMTR09I SJSD00 SYS1.PARMLIB
AMTROAI Current LNKLST set is IPL
AMT001A SYSPRG
```

**Legend:**

1. the message AMTR0PI is displayed only when a Program Event Recording (PER) SLIP trap is active

   Active PER SLIP traps have the potential to degrade processor performance so knowing that a PER SLIP trap is active is important.
Note
You can use the PER Exception Monitor sampler to notify you when a PER SLIP trap uses more than a user-specified percentage of CPU time. For more information, see “PER” on page 332.

2 system name obtained from field CVTSNAME
3 current MainView SYPROG Services release level
4 operating system name and level obtained from the CVT and ECVT
   Hiper Dispatch status obtained from the SVT.
5 CPU information obtained from the PCCA control blocks
6 LPAR information obtained via the STSI instruction
7 IPL information obtained from the ASMVT, SMCA, and IPA control blocks
8 TSO version level obtained from the TSVT and your terminal ID obtained from
   your TSB if applicable
9 status of the IBM common storage tracking facility
10 list of the datasets currently in the Logical PARMLIB concatenation
11 name of the LNKLST set currently in use for the system

IO

The IO service detects potential device performance delays by displaying all outstanding non-TP I/O.

Use this service to determine whether an address space is hung up because of an incomplete I/O operation, or to determine whether a device is in error recovery.

Syntax

```
IO
  dvn
  volser
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvn</td>
<td>device number for which outstanding non-TP I/O is to be displayed</td>
</tr>
<tr>
<td>volser</td>
<td>volume serial number for which outstanding non-TP I/O is to be displayed</td>
</tr>
</tbody>
</table>

**Note**
If you do not specify \( dvn \), the IO service monitors all outstanding non-TP I/O.

**Example 1**

To display all outstanding non-TP I/O, type

```
IO
```

Legend:

1. one-character flag; valid characters are as follows:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>line represents the I/O request currently active on the device</td>
</tr>
<tr>
<td>*</td>
<td>I/O request on this line has been queued by the I/O supervisor pending completion of a previous I/O request</td>
</tr>
<tr>
<td>+</td>
<td>ASM channel program is active on this device</td>
</tr>
</tbody>
</table>

2. address space name

3. device number

4. volume serial of the device, if it is a direct access storage or tape device

5. IOQ address

6. name of IOS driver; valid driver codes are as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISC</td>
<td>miscellaneous driver</td>
</tr>
<tr>
<td>EXCP</td>
<td>EXCP driver</td>
</tr>
<tr>
<td>VSAM</td>
<td>VSAM driver</td>
</tr>
</tbody>
</table>
7 seek address in CCHH format (DASD device only)
I/Os to the extended portion of Extended Addressability Volume (EAV) devices do not display the seek address.

Example 2

To display IO for volume PACK08, type

IO PACK08

AMTI11I GOJOB UNIT 283 PACK08 IOQ 00F83DA0 DRIVER-EXCP CCHH- 191 3 [1]
[2] [3] [4]
AMTI12I UNIT B00: RESERVES-0 ALLOCATIONS-3 OPENS-2

Legend:

1 device number, volume serial, IOQ and IOSB addresses of the I/O, the z/OS I/O driver, and the seek address for this I/O operation

2 number of outstanding RESERVE requests for this device
3 number of address spaces allocating this device
4 number of open DCBs against this device

**Note**
Message AMTI121 is displayed for DASD devices only.

---

**IPLDATA**

The IPLDATA (IPL) service displays information obtained from the Initialization Parameter Area (IPA), which is mapped by IBM macro IHAIPA.

**Note**
The IPL time, obtained from the IPA, can be several minutes after the start of system initialization.

---

**Syntax**

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>(optional) if specified, only the parameters that begin with the specified prefix will be displayed (see Example 2 on page 128)</td>
</tr>
</tbody>
</table>

For IEASYS parameters, the Source column indicates the source of the parameter and can be:

- the name of the parmlib member from which it was obtained
- OPER, indicating that it was specified by the operator

---

**Example 1**

To display all available IPL data, type

```bash
IPL
```

AMTIP1I LPAR=SJSD, IPL on Saturday, August 3, YYYY; 1:25:55.63 AM Local
AMTIP2I Loadparm Unit=A440, Volser=HCD002, SYS1.IPLPARM(LOADDO)
AMTIP0I
Example 2

You are interested in the values for parameters that begin with the letters GRS. To limit the display of parameter values to the ones of interest, type

IPL GRS

<table>
<thead>
<tr>
<th>Source Parm</th>
<th>IEASYSxx Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEASYSD1 GRS</td>
<td>TRYJOIN</td>
</tr>
<tr>
<td>Default GRSCNF</td>
<td>00</td>
</tr>
<tr>
<td>IEASYSD1 GRSRNL</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>IEASYSD1 CLPA</td>
<td>(4250,600M)</td>
</tr>
<tr>
<td>Default CVIO</td>
<td>ABOVE</td>
</tr>
<tr>
<td>IEASYSD1 Default DDM</td>
<td>00</td>
</tr>
</tbody>
</table>

Example 2

You are interested in the values for parameters that begin with the letters GRS. To limit the display of parameter values to the ones of interest, type

IPL GRS

<table>
<thead>
<tr>
<th>Source Parm</th>
<th>IEASYSxx Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEASYSD1 GRS</td>
<td>TRYJOIN</td>
</tr>
<tr>
<td>Default GRSCNF</td>
<td>00</td>
</tr>
<tr>
<td>IEASYSD1 GRSRNL</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>IEASYSD1 Default DDM</td>
<td>00</td>
</tr>
</tbody>
</table>

Example 2

You are interested in the values for parameters that begin with the letters GRS. To limit the display of parameter values to the ones of interest, type

IPL GRS

<table>
<thead>
<tr>
<th>Source Parm</th>
<th>IEASYSxx Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEASYSD1 GRS</td>
<td>TRYJOIN</td>
</tr>
<tr>
<td>Default GRSCNF</td>
<td>00</td>
</tr>
<tr>
<td>IEASYSD1 GRSRNL</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>IEASYSD1 Default DDM</td>
<td>00</td>
</tr>
</tbody>
</table>

Example 2

You are interested in the values for parameters that begin with the letters GRS. To limit the display of parameter values to the ones of interest, type

IPL GRS

<table>
<thead>
<tr>
<th>Source Parm</th>
<th>IEASYSxx Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEASYSD1 GRS</td>
<td>TRYJOIN</td>
</tr>
<tr>
<td>Default GRSCNF</td>
<td>00</td>
</tr>
<tr>
<td>IEASYSD1 GRSRNL</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>IEASYSD1 Default DDM</td>
<td>00</td>
</tr>
</tbody>
</table>

Example 2

You are interested in the values for parameters that begin with the letters GRS. To limit the display of parameter values to the ones of interest, type

IPL GRS

<table>
<thead>
<tr>
<th>Source Parm</th>
<th>IEASYSxx Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEASYSD1 GRS</td>
<td>TRYJOIN</td>
</tr>
<tr>
<td>Default GRSCNF</td>
<td>00</td>
</tr>
<tr>
<td>IEASYSD1 GRSRNL</td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>IEASYSD1 Default DDM</td>
<td>00</td>
</tr>
</tbody>
</table>
ISMF

The ISMF (ISM) service displays information about the recording of SMF data relative to the way it is being recorded.

Example 1 illustrates the data that is provided when the data is being recorded in SMF data sets. Example 2 illustrates the data that is provided when it is being recorded to the logstream.

Syntax

Example 1

To display SMF data, type

ISMF

```
AMTSI1I Records lost=               0
AMTSI2I Records written=      246,834
AMTSI3I Buffers written=       67,318
AMTSI4I Avg records/buffer=         3.67
AMTSI5I Buffer size=            4,096
AMTSI6I Full buffers=               0
AMTSI7I Maximum Buffers            672
AMTSI8I Pct Used  Total Blocks  Status  Data Set Name
AMTSI9I --------  ------------  ------  -------------
AMTSI9I   48.83%        15,000  Active  SYS1.MAN1
AMTSI9I    0.00%        15,000  Ready   SYS1.MAN2
AMTSI9I    0.00%        15,000  Ready   SYS1.MAN3
AMTSI0I
```

This example shows information displayed when SMF data is being recorded in a data set.

The buffer size is also the Control Interval (CI) size. The number of full buffers is usually zero and rarely more than one.

Example 2

To display SMF data, type

ISMF

```
AMTSICI Logstream Name=   IFASMF.SJSD.LOGR2
AMTSIDI SMF Record Types= 0:42, 44, 46, 50:51, 59:69, 80:98, 100:162,
AMTSIDD Status=           Connected, Active
AMTSIF1 Buffer Size=      65,532 bytes
AMTSIGI Last Write=        9:29:01.559   6/22/YYYY
AMTSI0I
AMTSICI Logstream Name=   IFASMF.SJSD.LOGR1
AMTSIDI SMF Record Types= 70:79, 240
AMTSIDD Status=           Connected, Active
AMTSIF1 Buffer Size=      65,532 bytes
```

Chapter 4  Services  129
This example shows information displayed when SMF data is being recorded in the logstream.

**JCPU**

The JCPU service provides the ability to locate the address spaces that are the highest users of CP, zIIP, and zAAP processors.

The JCPU service reports on the distribution of CP, zIIP, and zAAP processor utilization during the sample period. The information may be sorted by any column. When sorted by CP, zIIP, or ZAAP the column is sorted in descending sequence and the display terminates when the last non-zero value has been displayed in that column. The user may specify a sample period from 1-60 seconds. The default is five seconds.

The unused processor time during the interval is not measured. Therefore, the total utilization will always be 100% for the sorted column (subject to rounding errors).

**Syntax**

The parameters may be entered in any sequence.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>n</em></td>
<td>A numeric value is assumed to be the sample interval in seconds and must be 1-60. If a value greater than 60 is specified a warning message is issued and the duration is set to 60.</td>
</tr>
<tr>
<td><em>keyword</em></td>
<td>A non-numeric value must be one of the supported keywords and is used to identify the column used to sort the data. The keywords are: ASID, CP, NAME, ZAAP, and ZIIP. The default is ZIIP.</td>
</tr>
</tbody>
</table>
Example 1

This example shows sample processor usage for one second and sorts the resulting data by the column containing CP utilization in descending sequence.

```
JCPU,1,CP

<table>
<thead>
<tr>
<th>ASID</th>
<th>ASID</th>
<th>% CP</th>
<th>% zIIP</th>
<th>% zAAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>008A</td>
<td>BMRSLD1</td>
<td>50.95</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0076</td>
<td>BMVKEM11</td>
<td>46.76</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0006</td>
<td>XCFAS</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>008B</td>
<td>WLM</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00C2</td>
<td>BMVDWP3</td>
<td>0.15</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0069</td>
<td>DC$TCPBP</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0044</td>
<td>MIMGR</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>006E</td>
<td>DC$PAS</td>
<td>0.12</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0076</td>
<td>DC$CAS</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>009E</td>
<td>MSTRM</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0007</td>
<td>GRS</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>001C</td>
<td>JES2MON</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00A7</td>
<td>DCTDCAS</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0001</td>
<td>MASTER*</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0009</td>
<td>SMSVSAM</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00A4</td>
<td>IMWEBSRV</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00AD</td>
<td>MVQA@CAS</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00A5</td>
<td>TN3270</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0046</td>
<td>TRITUNE</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0028</td>
<td>MIMCON</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0065</td>
<td>CNMNETB</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0042</td>
<td>JES2S001</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>001E</td>
<td>JES2</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00A8</td>
<td>DCSMTC</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0068</td>
<td>ULTRAOPT</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0030</td>
<td>CTSGATE</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0033</td>
<td>RMF</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00EE</td>
<td>BSDPDP1</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0021</td>
<td>DCVBBIB</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0026</td>
<td>ZFS</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>00EA</td>
<td>CTSACS</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0014</td>
<td>RACF</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0043</td>
<td>CTSACD</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>004C</td>
<td>INTUNE</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0094</td>
<td>DEE7MSTR</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-----</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td>100.01</td>
<td>100.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

JITOKEN

The JITOKEN (JIT) service provides a method for uniquely referencing the same job instance over time.

Given a job name (or an ASID), the JITOKEN service returns a unique token that can be used in any subsequent service that accepts a job name as a parameter. Job Instance Tokens always begin with a colon (:)
Some SYSPROG Services users utilize MainView AutoOPERATOR or another automation product to help automate their operations. Occasionally, a user might need to determine whether a specific job is still executing. One method is to issue the STATUS command with the subject address space; this method produces a minimum amount of output—one line of output if the identified job exists, or a one-line message if the job is not found. However, the STATUS service might not locate the desired instance; therefore, you should use the JITOKEN service, instead.

### Syntax

```
JITOKEN — asi
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

### Example

To display the Job Instance Token and other information for XTST6PAS, type

```
JIT,XTST6PAS
```

```
AMTJT1I Jobname=XTST6PAS.PAS6 ASID=223 (X00DF) Token= :05SYMY4 [1]
AMTJT3I STEP=PAS6 STOKEN= 0000037C00000001 Owner=MVSSTC [2]
AMTJT5I Elapsed time=10:51:18.180 (in hours, minutes and seconds)[4]
AMTJT6I SrvClass=STCNRM [5]
AMTJTBI AcntData=
```

**Legend:**

1. information that identifies the job (displayed only if the address space is a started task, and the started task ID differs from the job name), as follows:
   - job name and started task ID, separated by a period
   - address space’s ASID in decimal format and hexadecimal format (the latter starting with X and enclosed in parentheses)
   - unique token assigned to this job instance

2. current step name, followed by the address space token, followed by the owner

3. day, date, and time when the job instance was initiated
The JSTORM service locates all address spaces that have acquired above-the-bar memory objects.

The current and high-water mark values for private, shared private, and common memory objects are collected for each address space. The results are displayed according to the sort order specified by the first (and only) parameter.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyword</td>
<td>represents one of the following eight columns of data that can be sorted: PRI PRIHW SHR SHRHW COM COMHW JOBNAME ASID</td>
</tr>
</tbody>
</table>

- Multiple keywords can be specified, but only the first one is accepted for the primary sort.

- The sort sequence for JOBNAME and ASID is ascending. All other keywords are descending.

- Address spaces that have a value of zero in the sorted column are not displayed.
The default sort order is JOBNAME.

Example

To sort on the SHR column, type

JST, SHR

<table>
<thead>
<tr>
<th>AMTJS1</th>
<th>AMTJS2</th>
<th>AMTJS3</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
<th>AMTJS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname</td>
<td>Hex</td>
<td>ASID</td>
<td>Current</td>
<td>High</td>
<td>Current</td>
<td>High</td>
<td>Current</td>
<td>High</td>
<td>Current</td>
<td>High</td>
<td>Current</td>
</tr>
<tr>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
</tr>
<tr>
<td>DC$TCPID</td>
<td>DC$TCPID</td>
<td>DC$TCPID</td>
<td>DC$TCPID</td>
<td>DC$TCPID</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
<td>DEENDIST</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
<tr>
<td>0092</td>
<td>0099</td>
<td>007C</td>
<td>0096</td>
<td>0081</td>
<td>0083</td>
<td>0096</td>
<td>0090</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
<td>0091</td>
</tr>
</tbody>
</table>

LABEL

The LABEL (LAB) service reads a tape label and displays its contents.

This service lets you find the data set name and other characteristics that are specified in the tape label.

Syntax

```
LABEL dvn [EXT]
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dvn</code></td>
<td>hexadecimal device number of the tape drive</td>
</tr>
<tr>
<td><code>EXT</code></td>
<td>causes a full 80 bytes of label information to be displayed</td>
</tr>
</tbody>
</table>

**Note**

An INTERVENTION REQUIRED message appears on the console for the tape drive on which the tape is to be mounted.
Example

To display the contents of the tape label on device 180, type

LABEL 180

<table>
<thead>
<tr>
<th>AMTL12I</th>
<th>VOL1000111</th>
<th>OWNRID [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTL12I</td>
<td>HDR1USER.DATASET 00000100010001 091775 [3]</td>
<td></td>
</tr>
<tr>
<td>AMTL12I</td>
<td>HDR2F035200008031MYJOB /STEPONE [4]</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1. volume label
2. data set name
3. header label #1
4. header label #2

LCPUS

The LCPUS (LCP) service provides the ability to display and modify the CPU time limits for an address space that were originally specified by the TIME parameter on the JOB and EXEC statements.

The CPU limit for the current job step and the limit for the job (maximum for all job steps) are displayed. The CPU time used and CPU time remaining for the job step are also displayed.

If new values are specified, the step limit is reset to the specified limit value. The job limit is also reset if the current value is less than the specified value; otherwise, it remains unchanged.

The maximum CPU limit value that can be specified is 1,440 minutes, which can be specified in minutes, seconds, or a combination of both.

Syntax

![Syntax Diagram]

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>minutes</td>
<td>number of minutes for the new CPU time limit The maximum value that can be specified is 1440.</td>
</tr>
<tr>
<td>seconds</td>
<td>number of seconds for the new CPU time limit You can specify the CPU time limit in minutes or seconds or both. If specifying in seconds only, you must insert two commas between the address space instance and the number of seconds, to indicate zero minutes.</td>
</tr>
<tr>
<td>U</td>
<td>in the third or fourth parameter position, indicates that the CPU time limit for the step can be set to a value less than the current limit value</td>
</tr>
</tbody>
</table>

### Examples

To display the current CPU time limits and the amount of time remaining for the job and job step, type

**LCPU, VAM3A**

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLC1I</td>
<td>CPU limit for job VAM3A</td>
<td>10 minutes</td>
</tr>
<tr>
<td>AMTLC2I</td>
<td>CPU limit for current step ASMSTEP</td>
<td>2 minutes, 30 seconds</td>
</tr>
<tr>
<td>AMTLC3I</td>
<td>Step time used</td>
<td>1 minutes, 35 seconds</td>
</tr>
<tr>
<td>AMTLC4I</td>
<td>Step time remaining</td>
<td>55 seconds</td>
</tr>
<tr>
<td>AMTLC0I</td>
<td>CPU limit changed [5]</td>
<td></td>
</tr>
</tbody>
</table>

To increase the CPU time limit for the current job step and to display the new limit values and remaining CPU time, type

**LCPU, VAM3A, 5**

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLC1I</td>
<td>CPU limit for job VAM3A</td>
<td>10 minutes</td>
</tr>
<tr>
<td>AMTLC2I</td>
<td>CPU limit for current step ASMSTEP</td>
<td>5 minutes</td>
</tr>
<tr>
<td>AMTLC3I</td>
<td>Step time used</td>
<td>1 minutes, 35 seconds</td>
</tr>
<tr>
<td>AMTLC4I</td>
<td>Step time remaining</td>
<td>3 minutes, 25 seconds</td>
</tr>
<tr>
<td>AMTLC0I</td>
<td>CPU limit changed [5]</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. message number
2. address space instance (asi)
3. minutes and seconds fields
4. CPU time limit message lines
5. notification of changed CPU limit
The LNKLST (LNK) service provides the ability to create, modify, and delete LNKLST sets.

It also provides the ability to

■ make a LNKLST set the active (current) set
■ switch address spaces to the current set
■ display related information

The LNKLST service will also search the user’s JOBLIB/STEPLIB, the LPA, and the current LNKLST set for a specified load module and display where it was found.

General information

The IBM Dynamic LNKLST facility introduced the concept of LNKLST sets. Each address space in the system is assigned to a LNKLST set. A LNKLST set defines the data sets that compose the LNKLST for the address spaces using that LNKLST set. LNKLST set names are one to 16 characters in length. The initial set is named IPL. The IPL set cannot be modified. New address spaces are assigned to the CURRENT set. The name CURRENT is generic and refers to the current LNKLST set, regardless of its actual name.

Here is a list of what you can and cannot do with LNKLST and its various functions:

■ You cannot modify or UNDEFINE a LNKLST set that has address spaces assigned to it.
■ You can use the LNKLST DEACT function or the LNKLST UPDATE function to switch address spaces to the current LNKLST set.
■ You can use the LNKLST ACTIVATE function to make the specified LNKLST set the current LNKLST set.
■ You can use the LNKLST PROCESS function to copy an existing LNKLST set, add data sets to an existing set, or delete data sets from an existing set.
■ You can use the LNKLST UNDEFINE function to remove an existing LNKLST set from the system.
■ You can use the functions STATUS and USERS to display information about the existing LNKLST sets and the address spaces assigned to them.
You can use the LOCATE function to search the data sets in the current LNKLST set for the specified load module and display the name of the data set containing the load module.

Syntax

### Note

All function keywords (first parameter) can be abbreviated to their first their letters.

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| STATUS    | displays a list of the defined LNKLST sets and the number of address spaces that are using each LNKLST set  
- A plus sign (+) preceding a set name indicates that it is the current LNKLST set.  
- A minus sign (-) preceding a set name indicates that the set had been the current set at one time. |
| LIST      | displays a list of the data sets in the specified LNKLST set and their sequence number, volser, APF, and SMS status  
If a set name is not specified, the current set is displayed. |
| USERS     | displays a list of address spaces that are currently assigned to the specified LNKLST set  
If a set name is not specified, the current set is displayed. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS</td>
<td>allows you to process a LNKLST set; you must specify at least one suffix</td>
</tr>
<tr>
<td>ACTIVATE</td>
<td>allows you to make the specified LNKLST the current set; you must specify a LNKLST set name</td>
</tr>
<tr>
<td>UPDATE</td>
<td>changes one or more address spaces to the current LNKLST set</td>
</tr>
<tr>
<td></td>
<td>Alternatively, you can specify ALL to indicate all address spaces or a jobname-pattern containing wildcard characters.*</td>
</tr>
<tr>
<td>DEACT</td>
<td>changes all address spaces assigned to the specified LNKLST set to the current LNKLST set</td>
</tr>
<tr>
<td>UNDEFINE</td>
<td>removes the specified LNKLST set from the system</td>
</tr>
<tr>
<td>LOCATE</td>
<td>searches the JOBLIB/STEPLIB, LPA, and the current LNKLST set for the specified load module and displays its location</td>
</tr>
<tr>
<td>CURRENT</td>
<td>displays currently active LNKLST set</td>
</tr>
<tr>
<td>suffix1, suffix2...suffixn</td>
<td>indicates that one or more member suffixes must be specified after the PROCESS keyword</td>
</tr>
<tr>
<td></td>
<td>(The suffixes are appended to PROG to form the member names to be processed.)</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>modname</td>
<td>limits the display to the specified module</td>
</tr>
</tbody>
</table>

**Note**

IBM permits the use of the question mark (?) and asterisk (*) to create a pattern to identify one or more address spaces. A question mark (?) indicates that any character in that position is to be accepted. An asterisk (*) indicates that zero to one character can be substituted.

For example, A*H indicates that all names that begin with the letter A and end with the letter H are to be accepted. A?H indicates that all three-character names that begin with an A and end with an H are accepted.

**Examples**

This section provides examples on how to use the LNKLST functions.

**LIST**

To display a list of the data sets in the IPL set, type

`LNK, LIST, IPL`

To display a list of the data sets in the current LNKLST set, type

`LNK, LIST`

*or*
**LNK, LIST, CURRENT**

**USERS**

The USERS function displays a list of the address spaces that are currently assigned to the specified LNKLST set. If a set name is not specified, information for the current LNKLST set is displayed. The information includes the set name, current number of address spaces, the ASID (in hexadecimal), and the job name for each address space.

To display a list of the address spaces that are using the IPL set, type the following command:

**LNK, USE, IPL**

To display a list of the address spaces that are using the current set, type the following command:

**LNK, USE**

**PROCESS**

The primary purpose of PROCESS is to create new LNKLST sets and modify existing sets. However, the functions UPDATE and UNDEFINE are supported as well.

PROCESS reads the specified PROGxx members and processes the following statements:

- LNKLST DEFINE
- LNKLST ADD
- LNKLST DELETE
- LNKLST UPDATE
- LNKLST UNDEFINE

Other statements (APF, EXIT SYSLIB, LNKLST TEST, and LNKLST ACTIVATE) are ignored.

Specify the suffixes of the PROGxx members that you want to process as the second parameter of the command. If you specify more than one suffix, you must enclose them in parentheses. For example, the following command will process members PROG01 and PROG03:

**LNKLST, PRO,(01,03)**

Creating a new LNKLST set or modifying an existing set is a two-step process.

- First, you need to create a PROGxx member in a data set included in the logical PARMLIB concatenation (formerly SYS1.PARMLIB).
- Second, use LNKLST, PROCESS to process the statements in PROGxx.
To identify the data sets that are to be included in the new or modified LNKLST set, use the PROG xx statements:

**LNKLST DEFINE**

**LNKLST ADD**

**LNKLST DELETE**

*Note*

If your PROGxx member contains symbols, they are replaced with their current value.

For example, assume that you want to add a new data set called SYS2.MYLIB to the beginning of the current LNKLST, which is the IPL LNKLST set. Because you cannot modify the IPL set, you create your own LNKLST set (called MYSET) by copying the IPL set, and then adding your data set to the copy.

First, create PARMLIB member PROGMY containing a LNKLST DEFINE statement with the COPYFROM parameter to define the new set, and then copy the data sets from the IPL set. Also include a LNKLST ADD statement to add your data set after SYS1.CSSLIB. Type

```
PROGMY
LNKLST define name(myset) copyfrom(ipl)
LNKLST add    name(myset) dsname(sys1.mylib) attop
```

*Note*

See the IBM publication, *OS/390 MVS Initialization and Tuning Reference*, for an explanation of the LNKLST statement used in member PROGxx.

Then use the following LNKLST command to process the member you have created:

**LNK,PRO,MY**

At this point, you have created LNKLST set MYSET. It is available now but will not be used until it has been activated. Once activated, new address spaces will be assigned to it, but the existing address spaces will continue to use the IPL set until you use either DEACT or UPDATE to switch address spaces to the current set.

If you want to have all address spaces use your new LNKLST set, you could execute the following commands to make MYSET the current set and transfer all address spaces that are using set IPL to the current set:

**LNK,ACT,MYSET**

**LNK,DEACT,IPL**

**ACTIVATE**

The ACTIVATE function makes the specified LNKLST set the current LNKLST set.

Type the following command to make LNKLST set MYSET the current LNKLST set:

**LNK,ACT,MYSET**
**UPDATE**

The UPDATE function switches one or more address spaces to the current LNKLST set. You can specify a job name or an ASID. If you specify a job name, you can use the wildcard characters * and ? to select a group of jobs. You can specify an ASID in hexadecimal or decimal format. Indicate that the value is an ASID by enclosing it in parentheses. Indicate that it is a hexadecimal value by prefixing it with an X. Therefore, (123) is a decimal ASID, and (X123) is a hexadecimal ID, equivalent to (291).

Type the following command to switch all address spaces to the current LNKLST set:

```
LNK,UPDATE,*
```

Type the following command to switch all jobs (and TSO users), starting with job names beginning with the letters ABC to the current set:

```
LNK,UPD,ABC*
```

Type the following command to switch all jobs (and TSO users), with ABC as the first three characters of the job name, any character in the fourth position, a 1 in the fifth position, and any characters in the remaining three positions:

```
LNK,UPD,ABC?1*
```

**DEACT**

The DEACT function switches all address spaces assigned to the specified LNKLST set to the current LNKLST set.

Type the following command to switch all address spaces assigned to LNKLST set MYSET to the current set:

```
LNK,DEACT,MYSET
```

**UNDEFINE**

The UNDEFINE function removes the specified LNKLST set from the system. A LNKLST set cannot be removed if any address spaces are using it. In addition, the IPL set cannot be removed.

*Note*

You can use DEACT to switch all of the address spaces to the current set.

Type the following command to remove LNKLST set MYSET from the system:

```
LNK,UND,MYSET
```

**LOCATE**

The LOCATE function searches the JOBLIB/STEPLIB, LPA, and current LNKLST set for the specified load module. If the module is found in the JOBLIB/STEPLIB concatenation, the concatenation number is displayed (the first data set is number 1).
If the load module is found in the current LNKLIST set, the name of the data set in which it was found is displayed.

LOADLPA

The LOADLPA (LOA) service lets you add modules or SVCs in LPA without needing to re-IPL.

This service also lets you delete modules that are added by this service.

Syntax

```
LOADLPA
  LIST
  ADD modname [USERLIB = DDname]
  DEL modname
  ALL
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| LIST      | lists information for all active modules  
|           | A module is considered active if it was loaded by the ADD function and has not been deleted. The LIST function is the default. |
| ADD       | loads a module and all of its aliases; it also places the module on the active LPA queue  
|           | Modules are loaded from the load library with a DD name of USERLIB, unless a data set name is specified as the third parameter.  
|           | If you specify a module that was previously loaded by the ADD function, the module is loaded at a new location because the entry point changes. The previously loaded copy is deleted (see the description of the DEL function). |
| modname   | name of the module that is to be added, deleted, or freed |
| dsname    | name of the data set where modname resides  
<p>|           | LOADLPA dynamically allocates the data set specified before loading modname into LPA. If no data set name is specified, LOADLPA tries to find modname in the DD concatenation USERLIB. Note that USERLIB must have been previously allocated to the same address space where SYSPROG Services is currently executing. (When SYSPROG Services is running under MainView for z/OS from the SYSPROG Services menu, this address space is PAS.) |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DEL       | deletes a module and its aliases loaded by the ADD function  
All saved entry-point addresses are restored, and the module is removed from the active LPA queue. The module’s storage is freed if the module’s use count is zero and if the module does not contain an SVC entry point.  
If you specify ALL, LOADLPA deletes all modules and aliases previously loaded by the ADD function. |

**Example 1**

To add the USERMOD module, which is found in LGS1.LOADLIB, type

LOADLPA ADD USERMOD LGS1.LOADLIB

| [2] AMTL5JI LA=00BA4100, LEN=0022CE |
| [3] AMTL5KI ALIAS=USEREP1, EP=00BA4140, CDE=F47B00 |
| [4] AMTL5KI ALIAS=USEREP2, EP=00BA4440, CDE=F47B30 |
| [5] AMTL5LI LOADLPA ADD FUNCTION COMPLETED |

**Legend:**

1 name of module loaded  
2 address of module entry point  
3 address of the contents directory entry (CDE) created for this module  
4 time module was loaded  
5 address of module  
6 length of module  
7 name, entry point address, and CDE address for all aliases found in the specified module for this module

**Example 2**

To delete the USERMOD module from LPA, type

LOADLPA DEL USERMOD

| AMTL5JII MODULE USERMOD DELETED |
| AMTL5PII LOADLPA DELETE FUNCTION COMPLETED |

**Usage notes**

- Entry points of some operating system routines are maintained in system control blocks. The entry points that are used for specific modules and control blocks
depend on the operating system, release level, and maintenance level. The LPA active queue is the only control block that LOADLPA updates. Therefore, programs that do not use LINK, LOAD, ATTACH, or XCTL to access the module are unaffected by LOADLPA.

- LOADLPA can delete only the module and aliases that LOADLPA has loaded.

- The SVC table is updated with an entry point address for any module name (true name or alias) that conforms to SVC naming conventions. If necessary, you can use the ZAP service to update the SVC table to change the locks. All types of extended SVCs are supported. In all cases, the old entry point address is saved and then restored when the module is deleted by the LOADLPA delete function. Use the ZAP service to reset the locks. See “SVCFIND” on page 249 and “ZAP” on page 277 for more information.

- Using LOADLPA, you can either add to or delete from LPA the SVC names that end with X'C0' (such as type 4 SVC 250) by substituting a left brace (\{) for the last characters of the module name; for example:

  LOADLPA ADD IGC0025{

- When you add an extended SVC with the LOADLPA service, the entry point address in the Extended SVC Router (ESR) table is updated. However, the flags in the second word of the ESR table entry are not changed. Use the ZAP service to reset flags, if required.

- Modules loaded by the LOADLPA service are logically added to the LPA but physically reside in CSA storage.

LOGGING

The LOGGING (LOG) service controls the logging system.

The logging system provides the ability to periodically execute a set of services with the output directed to a data set that you specified.

**Syntax**
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| xx        | suffix of the $$INLGxx member used to start logging  
The $$INLGxx member contains control statements that define the services to be executed and control the logging process, including the output data set. This parameter also implies start. |
| dsname    | name of the partitioned data set that contains the $$INLGxx member  
If this parameter is omitted, the name you last specified is used. If you did not previously specify a dsname, the data set allocated to DDNAME BBPARM is used. |
| STOP      | indicates that logging is to be stopped |
| FREE      | indicates that the logging output data set is to be freed; FREE implies stop |
| STATUS    | indicates that the status of the logging system is to be displayed |

### $$INLGxx control statements

The verb (statement type) must begin in column 1. One or more blanks are allowed between the verb and the operand. Operands cannot extend beyond column 72.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS c</td>
<td>indicates that the logging output is to be written to a SYSOUT data set of the specified class; CLASS is mutually exclusive with the DSN statement</td>
</tr>
</tbody>
</table>
| CMD command | command that is to be periodically executed  
Use multiple CMD statements for multiple commands. Commands are executed in the order that they appear. |
| CNT n     | specifies that logging is to be terminated when n number of cycles have been executed; also, see also STOP hhmm |
| DSN dsname| specifies that logging output is to be written to the preallocated data set dsname  
If no CLASS or DSN statement is provided, and a data set has been allocated to DDNAME LOG, the output will be written to that data set. Otherwise, it will be written to SYSOUT=A. |
<p>| EJC       | indicates the output produced by the command that follows should start on a new page; the EJC statement should be followed by one or more CMD statements |
| INCR n    | indicates the number (n) of seconds (increment) in an interval; default is 30 seconds; the product of INCR and INT determines the number of seconds between logging cycles |
| INT n     | indicates the number (n) of intervals in a logging cycle; default interval is one; the product of INCR and INT determines the number of seconds between logging cycles |</p>
<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START hhmm</td>
<td>indicates the time (hhmm) that logging is to start; specify 1300 for 1:00 P.M. Logging starts immediately if you do not provide a START statement.</td>
</tr>
<tr>
<td>STOP hhmm</td>
<td>indicates the time (hhmm) that logging is to terminate; specify 1200 for 12:00 P.M. (noon) and 2400 for 12:00 A.M. (midnight)</td>
</tr>
<tr>
<td></td>
<td>If you provide both CNT and STOP statements, logging will terminate when the first condition is reached.</td>
</tr>
<tr>
<td>HOLD</td>
<td>indicates that the SYSOUT output is to be held</td>
</tr>
<tr>
<td></td>
<td>If neither HOLD nor NOHOLD are specified, the default for the SYSOUT class will prevail. The HOLD statement is ignored if logging output is not written to a SYSOUT data set.</td>
</tr>
<tr>
<td>NOHOLD</td>
<td>indicates that the SYSOUT output is not to be held</td>
</tr>
<tr>
<td></td>
<td>If neither HOLD nor NOHOLD are specified, the default for the SYSOUT class will prevail. The NOHOLD statement is ignored if logging output is not written to a SYSOUT data set.</td>
</tr>
<tr>
<td>MAXOUT n</td>
<td>indicates the maximum number (n) of lines that will be written to a SYSOUT data set</td>
</tr>
<tr>
<td></td>
<td>Logging will be terminated when n lines have been written to SYSOUT. You can specify up to eight digits. The default is to write an infinite number of lines. MAXOUT is only applicable to SYSOUT data sets.</td>
</tr>
</tbody>
</table>

**Examples**

**Member $$INLG01**

```
INCR 60
INT 5
DSN VAM3.LOG
START 2200
STOP 2259
CMD CSA,MAP
```

Member $$INLG01$$ will cause the command CSA,MAP to be executed once every five minutes between 10:00 P.M. and 10:59 P.M. The output will be written to data set VAM3.LOG.

**Member $$INLG02$$**

```
INCR 30
INT 2
CNT 60
HOLD
CLASS A
CMD MTP
CMD ENQ
CMD CPU 5
```

Member $$INLG02$$ causes the commands MTP, ENQ, and CPU 5 to be executed once a minute for a total of 60 times. The output will be written to SYSOUT Class A and placed on the HOLD queue.
LWAIT

The LWAIT (LWA) service displays the wait limit for an address space.

If the address space is currently waiting, LWAIT also displays the amount of time it has been waiting since it was last dispatched, and the remaining wait time available before the address space will be terminated by the operating system with a 522 abend.

In addition, the LWAIT service provides the ability to increase or decrease the wait limit for the address space. To prevent errors, you must also specify U to decrease the wait limit.

Note

The wait limit is the amount of time the address space can remain in a continuous wait.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>hours</td>
<td>number of hours for the new wait limit</td>
</tr>
<tr>
<td>minutes</td>
<td>number of minutes for the new wait limit</td>
</tr>
<tr>
<td></td>
<td>You can specify the wait limit in hours and minutes or minutes only. If specifying in minutes only, you must insert two commas between the address space instance and the number of minutes to indicate zero hours.</td>
</tr>
<tr>
<td>U</td>
<td>in the third or fourth parameter position, indicates that the wait limit can be set to a value less than the current limit value</td>
</tr>
</tbody>
</table>

Examples

To display the wait limit and the amount of time the job has been in a wait state, type
LWAIT,VAM3W1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLW1</td>
<td>VAM3W1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I</td>
<td>Has been waiting for [6]</td>
<td>4 minutes, 37 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I</td>
<td>Wait limit remaining [6]</td>
<td>1 hours, 55 minutes, 22 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I</td>
<td>Wait limit is [6]</td>
<td>2 hours, 00 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLWoI</td>
<td>Wait limit changed [7]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To increase the wait limit to three hours, type this command and the new wait limit is displayed:

**LWAIT,VAM3W1,3**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLW1</td>
<td>VAM3W1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I</td>
<td>Has been waiting for [6]</td>
<td>4 minutes, 58 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I</td>
<td>Wait limit remaining [6]</td>
<td>2 hours, 55 minutes, 22 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I</td>
<td>Wait limit is [6]</td>
<td>3 hours, 00 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL WoI</td>
<td>Wait limit changed [7]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The request to set the wait limit to 30 minutes was denied because the current wait limit is a larger value (one hour) and U was not specified.

**LWAIT,VAM3W1,30**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLW1</td>
<td>Request denied. New value is less than old value.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW1</td>
<td>VAM3W1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I</td>
<td>Has been waiting for [6]</td>
<td>6 minutes, 45 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I</td>
<td>Wait limit remaining [6]</td>
<td>53 minutes, 14 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I</td>
<td>Wait limit is [6]</td>
<td>1 hour, 00 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL WoI</td>
<td>Wait limit changed [7]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The wait limit is reduced to 30 minutes, as requested. Note that U was specified to allow the reduction.

**LWAIT,VAM3W1,30,U**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLW1</td>
<td>VAM3W1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I</td>
<td>Has been waiting for [6]</td>
<td>7 minutes, 10 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I</td>
<td>Wait limit remaining [6]</td>
<td>22 minutes, 49 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I</td>
<td>Wait limit is [6]</td>
<td>30 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL WoI</td>
<td>Wait limit changed [7]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1 message number

2 address space instance (asi)

3 hours field

4 minutes field

5 seconds field

6 wait limit message lines

7 notification of changed wait limit
The LX service displays information for system and nonsystem linkage indexes (LXs).

Depending on the parameter value, the LX service displays summary information, information for a specific LX number, information for a specific address space, or a list of all LXs.

**Syntax**

```
LX [asi] [lx number] ALL
```

**Note**

When the service name only is specified (Example 2 on page 151), the LX service displays summary information consisting of:

- status of the related architecture features
- number of system and nonsystem standard LXs; number of system and nonsystem big LXs
- list of dormant LXs containing the job name and ASID of the address spaces that allocated them
- list of the top-ten LX owners

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>lx number</td>
<td>LX number, which is a decimal or hexadecimal value Hexadecimal values should be in the form x'value'. For example, the hexadecimal value ABC should be specified as X'ABC'. Values not expressed in that manner that do not begin with a numeric value will be assumed to be a job name. If an address space with that name does not exist and the value is a valid hexadecimal value, it will be assumed to be an LX number in hexadecimal. Decimal values will always be treated as a decimal LX number.</td>
</tr>
</tbody>
</table>
**Example 1**

To display LX information for job name VLF, type

**LX, VLF**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>keyword indicating that a display of all LXs is desired</td>
</tr>
</tbody>
</table>

See Example 2 on page 151.

---

**Example 2**

To display summary information, type

**LX**

```plaintext
AMTLXCI  ASN and LX Reuse architecture installed
AMTLXDI  ASN and LX Reuse architecture is enabled
AMTLXOI  
AMTLX2I Description               Used   Max  Pct Used
AMTLXEI  ------------------ -------- -------- -------- 
AMTLXFI  System Standard           180    223    80.71%
AMTLXFI  Non-Sys Standard          34     1,824   1.86%
AMTLXFI  System Big                133    1,024   12.98%
AMTLXFI  Non-Sys Big               114    29,697  0.38%
AMTLXOI  
AMTLX7I  LX Num Flags ASID Jobname LX Num Flags ASID Jobname
AMTLX8I  ------ ------- --------      ------ ------- -------- 
AMTLX9I  00000A  001E VLF
AMT001A SYSPROG

AMTLX3I  Dormant System LX's -
AMTLX4I  LX Num  Jobname   ASID
AMTLX5I  ------  --------  ----
AMTLX6I  000018  HWIBCPII  0019
AMTLX6I  00003F  MTACASDT  0078
AMTLX6I  000041  DHW1IRLM  00A1
AMTLX6I  000044  MTALOGRD  0088
AMTLXOI  
AMTLXKI  Top Ten LX Owners ---
AMTLX6I  Job Name   Number of LXs
AMTLXHI  --------   ------------
AMTLXJI  *MASTER*       105
AMTLXJI  DEEOMSTR       13
AMTLXJI  DHH1MSTR       13
AMTLXJI  DIB1MSTR       13
AMTLXJI  D12MSTR        13
AMTLXJI  EYUX410        9
AMTLXJI  CSO7MSTR       6
AMTLXJI  CSIBMSTR       6
AMTLXJI  DE00BM1        6
AMTLXJI  DHH1BM1        6
```
Example 3

To display information for LX number 10 decimal, type

**LX,10**

<table>
<thead>
<tr>
<th>AMTLXCI</th>
<th>Description</th>
<th>Used</th>
<th>Max</th>
<th>Pct Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXDI</td>
<td>System</td>
<td>180</td>
<td>223</td>
<td>80.71%</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Standard</td>
<td>34</td>
<td>1,824</td>
<td>1.86%</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System</td>
<td>133</td>
<td>1,024</td>
<td>12.98%</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Big</td>
<td>114</td>
<td>29,697</td>
<td>0.38%</td>
</tr>
</tbody>
</table>
monitors the I/O activity of all devices and address spaces (it can also monitor the I/O activity of a specified device or address space)

detects I/O bottlenecks that cause workload delays

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvn</td>
<td>device number</td>
</tr>
<tr>
<td>dvn1-dvn2</td>
<td>range of device numbers; dvn1 is the lowest device number and dvn2 is the highest device number</td>
</tr>
<tr>
<td>time</td>
<td>specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum permitted value is 600 seconds (10 minutes).</td>
</tr>
</tbody>
</table>

The value is calculated as follows:

ServiceTime = PendingTime + DoscpmnectTime + ConnectTime

The value is averaged over the interval specified in the MDEV command (in this example, 15 seconds). It is shown in milliseconds (ms).

**Examples**

To monitor all devices for 15 seconds, type

MDEVICE

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>944</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3375</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8362</td>
<td>BAB313</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8875</td>
<td>.26</td>
<td>.1</td>
<td>0</td>
<td>0</td>
<td>.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 4 Services 153
<table>
<thead>
<tr>
<th>AMTO38I</th>
<th>MTA9BMP4 017C</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO38I</td>
<td>MTA8BMP4 016D</td>
<td>12</td>
</tr>
<tr>
<td>AMTO38I</td>
<td>MTABMSG 0174</td>
<td>12</td>
</tr>
<tr>
<td>AMTO36D</td>
<td>--------------</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1. device number whose activity was measured
2. volume serial number on the device measured
3. percentage of time an I/O request was active for the device
4. average number of I/O requests that were queued by the z/OS/XA I/O supervisor for the device
5. total number of head movements observed for the device during the interval (N/A for non-DASD and fixed-head devices)
6. average number of cylinders moved for each seek observed for the device
7. number of I/O requests per second addressed to this device
8. average time (in milliseconds) that the device was connected to a channel path on a per-request basis
9. average time (in milliseconds) that each request was held, pending availability of a wait for path, device, or control unit
10. average time (in milliseconds) that the device was disconnected from the channel path pending a device or control unit function (seek) on a per-request basis
11. DAO -- Device Active Only - average time (in milliseconds) that the device was device active, but not sub-channel active
12. The service time provides an indication of the average time it takes to service a I/O request. Service time for a device should be less than one revolution of the disk (around 8ms for many but not all devices). If it exceeds this there may be a problem. The user should decide on a threshold value for their I/O performance goals and if this SvTime value exceeds their threshold, they should investigate the device, and perhaps move some of the datasets on the volume to another device to balance the load.
13 name of an address space found to be a significant user of the device In this field, ****** indicates that the address space terminated between the time that the data was collected and the time that the MDEVICE report was generated. *OTHERS* in this field indicates that the address space had more users during the monitoring period than could be listed. All activity for the overflow of users is reported as *OTHERS*.

14 address space ID found to be a significant user of the device

15 percentage of the device busy time that this address space was requesting the device

**Usage note**

The MDEVICE service uses a sampling technique for measuring I/O activity. The default sampling period is 15 seconds at a rate of four samples per second. If you enter an override value for time, the sample rate is four samples per second if the period is 15 seconds or less and two samples per second if the period is greater than 15 seconds.

**Note**
The seek and cylinder information is not provided for I/O to the extended portion of Extended Addressability Volume (EAV) devices.

**MEMSCAN**

The MEMSCAN (MEM) service scans the specified storage areas for the specified character or hexadecimal string and displays the location of each occurrence of the string.

In addition, a symbol of the form @nn is created for each string occurrence until 99 symbols have been created. You can use these symbols in conjunction with the DUMP and ZAP services to display or modify the located data.

When 99 symbols have been created, MEMSCAN displays 64 bytes of storage containing the next occurrence of the string and then terminates. You can resume the scan to find the next occurrence by typing MEMSCAN without any operands. You can use an asterisk (*) to reference the address of the last occurrence of the string symbolically on the DUMP, EQUATE, and ZAP services.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>displays all character strings that are to be interpreted as uppercase and do not contain any special characters. Enclose the value in single quotation marks if it contains lowercase or special characters. For example, ‘ABC,def’ is a seven-character string that begins with three uppercase characters, followed by a special character and three lowercase characters. Characters that are typed in lowercase but not enclosed in single quotation marks are converted to uppercase.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| X'string' | displays hexadecimal data  
          | Hexadecimal data must be enclosed in single quotation marks and preceded by an X; for example: X'C1C2C36BC4C5C6'. |
| ?        | displays a map of storage depicting the various areas within the address space |
| @        | displays the symbols previously created by MEMSCAN |
| *        | represents the address of the last location set by one of several services  
          | Typing an asterisk is equivalent to entering the address as a hexadecimal string. See Example 1 on page 158. |
| Null     | omitting all parameters or omitting the first parameter causes MEMSCAN to resume an interrupted search |
| area     | subsequent parameters (2nd through 10th) as described in the “Area keywords” on page 157 section |
| RFIND    | finds a previous scan |

---

**Note**

You can specify up to 10 parameters.

---

**Area keywords**

Any of the following keywords can be specified in any sequence, separated by commas or blanks. Only the first three letters of each keyword are required. Area keywords are cumulative in their effect. If an area keyword is not specified, the default is COMMON.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOVE</td>
<td>causes any previously created symbols of the format @nn to be deleted prior to starting the scan (where nn is 01 through 99)</td>
</tr>
<tr>
<td>DUMP</td>
<td>causes MEMSCAN to display 64 bytes of storage surrounding each occurrence of the string</td>
</tr>
<tr>
<td>ABOVE</td>
<td>scans all storage above the 16-megabyte line</td>
</tr>
<tr>
<td>ALL</td>
<td>scans all storage (0 through 7FFFFFFF)</td>
</tr>
<tr>
<td>BELOW</td>
<td>scans storage below the 16-megabyte line</td>
</tr>
<tr>
<td>COMMON</td>
<td>scans common storage, which comprises CSA, ECSA, SQA, and ESQA; the PSA, LPA, and nucleus area, although in common storage, are excluded</td>
</tr>
<tr>
<td>CSA</td>
<td>scans the Common Service Area (CSA) located below the 16-megabyte line</td>
</tr>
<tr>
<td>ECSA</td>
<td>scans the Extended Common Service Area</td>
</tr>
<tr>
<td>EFLPA</td>
<td>scans the Extended Fixed Link-Pack Area</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EMLPA</td>
<td>scans the Extended Modified Link-Pack Area</td>
</tr>
<tr>
<td>EPLPA</td>
<td>scans the Extended Pageable Link-Pack Area</td>
</tr>
<tr>
<td>EPVT</td>
<td>scans the Extended Private area</td>
</tr>
<tr>
<td></td>
<td>Scans only the private area of the address space within which the SYSPROG</td>
</tr>
<tr>
<td></td>
<td>Services service is executing.</td>
</tr>
<tr>
<td>ERONUC</td>
<td>scans the Extended read-only nucleus</td>
</tr>
<tr>
<td>ERWNUC</td>
<td>scans the Extended read/write nucleus</td>
</tr>
<tr>
<td>ESQA</td>
<td>scans the Extended System Queue Area</td>
</tr>
<tr>
<td>FLPA</td>
<td>scans the Fixed Link-Pack Area</td>
</tr>
<tr>
<td>LPA</td>
<td>scans the entire Link-Pack Area, including FLPA, MLPA, PLPA, EFLPA, EMLPA,</td>
</tr>
<tr>
<td></td>
<td>and EPLPA</td>
</tr>
<tr>
<td>MLPA</td>
<td>scans the Modified Link-Pack Area (below the 16-megabyte line)</td>
</tr>
<tr>
<td>NUCLEUS</td>
<td>scans the entire nucleus, including RWNUC, RONUC, ERWNUC, and ERONUC</td>
</tr>
<tr>
<td>PLPA</td>
<td>scans the Pageable Link-Pack Area (below the 16-megabyte line)</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>scans private storage, both below and above the 16-megabyte line</td>
</tr>
<tr>
<td>PSA</td>
<td>scans the Prefixed Save Area</td>
</tr>
<tr>
<td>PVT</td>
<td>scans private storage below the 16-megabyte line</td>
</tr>
<tr>
<td>RONUC</td>
<td>scans the read-only nucleus below the 16-megabyte line</td>
</tr>
<tr>
<td>RWNUC</td>
<td>scans the read/write nucleus below the 16-megabyte line</td>
</tr>
<tr>
<td>SQA</td>
<td>scans the System Queue Area below the 16-megabyte line</td>
</tr>
</tbody>
</table>

### Example 1

To locate vectors to IGGPOST0, type **LPA,IGGPOST0**

```
AMTL41I MODULE=IGGPOST0 < EPLPA >          [1]
AMTL42I LA=030ADF38  EP=00AF72F8 (AMODE-31) LEN=000008
AMTL43I LPDE=00C435B8
AMTL44I SYSPROG MEMSCAN,*
AMTVS8I SEARCHING FOR X' '                  [2]
AMTVS8S WILL BE SCANNING: CSA 00800000-00B97FFF
AMTVS8S WILL BE SCANNING: SQA 00EBE000-00FC9FFF
AMTVS8S WILL BE SCANNING: EXT-SQA 016D2000-020AFFFF
AMTVS8S WILL BE SCANNING: EXT-CSA 043D4000-07FFFFFF
AMTVS8N SCANNING: CSA 00800000-00B97FFF
AMTVS8N SCANNING: SQA 00EBE000-00FC9FFF
AMTVS8I STRING FOUND AT 00C435AC IN SQA @01   [3]
AMTVS8N SCANNING: EXT-SQA 016D2000-020AFFFF
AMTVS8N SCANNING: EXT-CSA 043D4000-07FFFFFF
AMTVS8S SEARCH STOPPED AT THE END OF THE AREA (07FFFFFF)
AMTL001A SYSPROG DUMP,@01-20
```

158  MainView SYSPROG Services User Guide and Reference
Legend:

1. First, the LPA service is used to locate the start of IGGPOST0.
2. Then, MEMSCAN is used to search common storage for the address of IGGPOST0.
3. Finally, the DUMP service is used to display the storage area containing the located address constant.

**Example 2**

To produce a memory map, type

MEM,?

<table>
<thead>
<tr>
<th>AMTVSH1</th>
<th>ALLOCATED STORAGE AREAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVSH2</td>
<td>AREA TITLE</td>
</tr>
<tr>
<td>AMTVSLN</td>
<td>-------------------------</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>PSA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>PRIVATE</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>CSA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>MLPA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>PLPA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>SQA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>NUC R/W</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>NUC R/O</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-NUC R/O</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-NUC R/W</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-SQA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-PLPA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-FLPA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-MLPA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-CSA</td>
</tr>
<tr>
<td>AMTVSAI</td>
<td>EXT-PRIVATE</td>
</tr>
</tbody>
</table>

**Example 3**

To scan all storage for a hexadecimal string and display each occurrence, type

MEM,X’00AF72F8’,DUMP,ALL

<table>
<thead>
<tr>
<th>AMTVSAI</th>
<th>SEARCHING FOR X’00AF72F8’</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVWS</td>
<td>WILL BE SCANNING: ALL STORAGE</td>
</tr>
<tr>
<td>AMTVSN</td>
<td>SCANNING: PRIVATE 00010000-007FFFFFF</td>
</tr>
<tr>
<td>AMTVSN</td>
<td>SCANNING: CSA 00800000-00B97FFF</td>
</tr>
<tr>
<td>AMTVSN</td>
<td>STRING FOUND AT 00AF7384 IN CSA @01 AMTVSLN</td>
</tr>
<tr>
<td>AMTVSN</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>00AF7380 +04 F10000B0 00AF72F8 00AF73BB 00AFAF80 *1 8 *</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>00AF7390 00000000 00000000 00000000 00000000 *</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>00AF73A0 00000000 00000000 00000000 00000000 *</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>00AF73B0 00000000 00000000 00000000 00AF9140 0751FE40 *</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>00AF7380 00B98000-00C2FFFF</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: MLPA 00C30000-00EB000</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: SQA 00EBE000-00FC9FFF</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: NUC R/W 00FCA000-00FDF000</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: NUC R/O 00FDA000-00FFFFFFFF</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: EXT-NUC R/O 01000000-01466FFF</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>SCANNING: EXT-NUC R/W 01467000-016D1FFF</td>
</tr>
<tr>
<td>AMTVSV</td>
<td>STRING FOUND AT 01B2E0AC IN EXT-SQA @02</td>
</tr>
</tbody>
</table>

Chapter 4 Services 159
Usage notes

- You can omit previously typed parameters if you do not want to change them. For example, if you had previously typed MEMSCAN TEXT PRIVATE (which searches for the string TEXT in private storage), you can search for the string TEXT in common storage by typing MEMSCAN,COMMON.

- The ATTENTION key interrupts MEMSCAN and saves all current values. The scan can be resumed at the point of interruption by typing MEMSCAN REFIND or just MEMSCAN.

- Do not embed blanks in the search string.

MIO

The MIO service monitors the overall performance of the entire I/O subsystem, or a subset thereof, to detect I/O bottlenecks causing workload delays.

This service provides workload (address space-related) information, which lets you detect jobs monopolizing specific devices and channel paths.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chpid</td>
<td>channel path identifier</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>chpid1-chpid2</td>
<td>range of channel path identifiers; chpid1 is the lowest channel path identifier, and chpid2 is the highest channel path identifier</td>
</tr>
<tr>
<td>time</td>
<td>specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 1,200 seconds</td>
</tr>
</tbody>
</table>

**Note**

If you do not specify `chpid`, the MIO service monitors all channel paths attached to all processors.

**Examples**

To display information about all channel paths, enter:

```
MIO
```

```
AMTO30I I/O data being gathered for 15 seconds
```

```
AMTO31I Channel Path Utilization Report
```

```
AMTO32I CHANNEL PATH 00 (BYTE) WAS 5% BUSY
AMTO32I CHANNEL PATH 03 (BLOCK) WAS 23% BUSY
AMTO32I CHANNEL PATH 12 (BLOCK) WAS 45% BUSY
AMTO32I CHANNEL PATH 23 (BLOCK) WAS 17% BUSY
```

```
AMTO33I Processor I/O Interrupt Activity Report
```

```
AMTO34I CPU 00 Processed 4,670 Interrupts ( 6% were through TPI)
```

```
AMTO35I Device and Logical Control Unit Report
```

```
AMTO36D -----------------------------------------------------
AMTO36H  Dev Volser %Bsy Qlth Seek ACYL Rate Conn Pend Disct DAO
AMTO36I 3500            3 0 0 0
AMTO38D            ------------------
AMTO38H            ASIDname ASID %Bsy
AMTO38I            CONSOLE 000A 100
AMTO36D ---------------------------------------------------------
AMTO36H  Dev Volser %Bsy Qlth Seek ACYL Rate Conn Pend Disct DAO
AMTO36I 858E USSA01    0 0 0 0 .20 0 0     0     0
AMTO38D            ------------------
AMTO38H            ASIDname ASID %Bsy
AMTO38I            MV60DS36 014F 100
AMTO36D ----------------------------------------------------------
AMTO36H  Dev Volser %Bsy Qlth Seek ACYL Rate Conn Pend Disct DAO
AMTO36I 8370 BAB327    7 0 2 1 25 4 .20 0 0
AMTO38D            ------------------
AMTO38H            ASIDname ASID %Bsy
AMTO38I            AAONS73 012A 40
AMTO38I            AAOD72B 00A6 60
```

**Legend:**

1. channel path identifier (CHPID)

2. type of channel path as specified during IOCP generation

   The valid path types are BLOCK, for block multiplexing paths, and BYTE, for byte multiplexing paths.

3. percentage of time that the channel path was busy during the sample period
Note

Channel paths less than 1% busy during the sample period are not reported unless specified either explicitly or implicitly (using a path range) as an operand of the MIO service.

4 CPU identifier (CPUID) for the processor whose activity is being reported

5 total number of I/O interrupts (including TPIs) processed by this CPU in the sample period

6 percentage of I/O interrupts processed by this CPU that did not require a physical PSW swap (that is, were processed through the Test Pending Interrupt (TPI) instruction)

7 device number whose activity was measured

8 volume serial number of the device measured

9 percentage of time that an I/O request was active for the device

10 average number of I/O requests queued by the I/O supervisor for the device

11 total number of head movements observed for the device during the sample period (N/A for non-DASD and fixed-head devices)

12 average number of cylinders moved for each seek observed for the device

13 number of I/O requests per second addressed to this device

14 average time (in milliseconds) that the device was connected to a channel path on a per-request basis

15 average time (in milliseconds) that each request was held, pending availability of a waiting for path, device, or control unit

16 average time (in milliseconds) that the device was disconnected from the channel path pending a device or control unit function (seek) on a per-request basis

17 name of an address space found to be a significant user of the device

In this field, asterisks (*********) indicates that the address space terminated between the time that the data was collected and the time that the MDEVICE report was generated. The value *OTHERS* in this field indicates that the address space had more users during the sample period than could be listed. All activity for the overflow of users is reported as *OTHERS*.

18 number of the address space found to be a significant user of the device
19 percentage of the device busy time that this address space was requesting the device

**MLCU**

The MLCU (MLC) service monitors the performance and utilization of logical control units configured into the I/O subsystem.

The service displays data (I/O rate, requests deferred for control unit, and device busy conditions) for logical control units that have I/O activity during the monitoring period.

Use this service to determine the optimal physical device placement and to evaluate shared DASD contention.

**Syntax**

![Syntax Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lcu</em></td>
<td>logical control unit</td>
</tr>
<tr>
<td><em>lcu1-lcu2</em></td>
<td>range of logical control units; lcu1 is the lowest-numbered logical control unit, and lcu2 is the highest-numbered logical control unit</td>
</tr>
<tr>
<td><em>time</em></td>
<td>specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 1,200 seconds</td>
</tr>
</tbody>
</table>

**Example**

To monitor all LCUs for 15 seconds, type

```
MLCU
```

```
AMTO40I LOGICAL Control UNIT DATA BEING GATHERED
AMTO41I LOGICAL Control UNIT ACTIVITY REPORT
[1] \[2]  
AMTO63I IOP 01: ACTIVITY RATE 0000 AVG Q LENGTH 0000  
AMTO611 LCU 023: CONTENTION RATE 0005 ALL PATHS BUSY 0001
```

Chapter 4  Services  163
Legend:

1. total number of I/O requests per second received for all devices associated with the LCU
2. average depth of the I/O request queue
3. rate at which I/Os were queued on the LCU in the I/O subsystem
4. percentage of time that all paths to a given I/O device on that LCU were busy
5. average length of the queue for the LCU
6. percentage of I/O requests processed by this LCU that were initiated successfully
7. percentage of I/O requests processed by this LCU that were deferred because of a busy physical control unit

Usage note

The MLCU service uses a sampling technique for measuring I/O activity. The default sampling period is 15 seconds at a rate of four samples per second. If you enter an override value for time, the sample rate is four samples per second if the period is 15 seconds or less and two samples per second if the period is greater than 15 seconds.

MONITOR

The MONITOR (MON) service monitors address space activity to determine whether an address space is in a wait or a looping state.

The service provides the basis for a deadline management system by providing CPU time and EXCP counts for the life of a step. For a job whose characteristics are known, you can gauge the progress of a step at any point during its execution.

Syntax

```
MONITOR asi [seconds]
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>seconds</td>
<td>number of seconds that the address space is to be monitored; the default is 10 seconds, and the maximum is 1,200 seconds</td>
</tr>
</tbody>
</table>

**Example**

To monitor address space XTSTQPAS for 10 seconds, type

```
MONITOR,XTSTQPAS
```

<table>
<thead>
<tr>
<th>AMTJ22I</th>
<th>Statistics being gathered for XTSTQPAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTJ211</td>
<td>STC07754 XTSTQPAS XTSTQPAS, Service Class=STCNRM</td>
</tr>
<tr>
<td>AMTJ231</td>
<td>Life of step CPU 305.91 Excp 529156 Pages 8410 SUs 6M</td>
</tr>
<tr>
<td>AMTJ241</td>
<td>Last 10 Secs CPU 0.03 Excp 0 Pages 0 SUs 670</td>
</tr>
</tbody>
</table>

**Legend:**

1. job ID
2. job name
3. step name
4. service class
5. CPU time for the life of the step and during the specified time
6. count of EXCPs for the life of the step and during the specified time
7. total number of page actions for the life of the step and during the specified time
8. total service units for the life of the step and during the specified time

**MPATH**

The MPATH (MPA) service monitors I/O channel path activity.

This service also detects imbalances and possible path failures by monitoring the utilization of all, or a subset of, the channel paths installed in the system.
## Syntax

![Diagram showing MPATH command with parameters chpid, range of chpid1-chpid2, and time.

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chpid</td>
<td>channel path identifier</td>
</tr>
<tr>
<td>chpid1-chpid2</td>
<td>range of channel path identifiers; chpid1 is the lowest channel path identifier, and chpid2 is the highest channel path identifier</td>
</tr>
<tr>
<td>time</td>
<td>specifies the length of the sample period in seconds; the default sample period is 15 seconds</td>
</tr>
</tbody>
</table>

**Note**

If you do not specify `chpid`, the MPATH service monitors all channel paths attached to all processors.

### Example 1

To display I/O activity for all channel paths, type

```plaintext
MPATH
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO2I CHANNEL PATH 0C WAS 4% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 12 WAS 1% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 13 WAS 1% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 1B WAS 13% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 1C WAS 7% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 38 WAS 14% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 39 WAS 19% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 3C WAS 19% BUSY</td>
<td></td>
</tr>
</tbody>
</table>

### Example 2

To display I/O activity for channel paths 12-14, type

```plaintext
MPATH 12-14
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO2I CHANNEL PATH 12 WAS 4% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 13 WAS 3% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO2I CHANNEL PATH 14 WAS 0% BUSY</td>
<td></td>
</tr>
</tbody>
</table>
Example 3

To display I/O activity for all channel paths over the past 60 seconds, type

MPA,60

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO21I CHANNEL PATH 0C WAS 4% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 12 WAS 3% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 13 WAS 3% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 14 WAS 4% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 1B WAS 5% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 1C WAS 5% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 1E WAS 4% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 3B WAS 5% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 39 WAS 11% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 3C WAS 11% BUSY</td>
<td></td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 3D WA 6% BUSY</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1  channel path identifier (CHPID)

2  percentage of time that the channel path was busy during the sample period

MSTORAGE

The MSTORAGE (MST) service maps virtual storage areas above and below the 16 MB line, including location, length, and percentage used, when applicable.

Areas common to all address spaces and private area locations specific to the specified address space are displayed, when available.

The private area above and below the 16 MB line is divided into three sections:

- LSQA/SWA--allocated from the top of the private area down
- User-committed private area--allocated from the bottom up
- Uncommitted area--between the LSQA/SWA and user-committed private area

The current utilization within the LSQA/SWA and committed private area is displayed. LSQA/SWA can expand into the uncommitted area as needed. Committed private area can expand into the uncommitted private area to the user limit value.

Syntax

MSTORAGE  asi
The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

**Example**

To display the virtual storage areas for BMVDWP3, type

```
MSTORAGE,BMVDWP3
```

<table>
<thead>
<tr>
<th>AMTMS1I AREA [2]</th>
<th>START</th>
<th>END</th>
<th>LENGTH (K)</th>
<th>%USED</th>
<th>USER LMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTMS31 Ext LSQA/SWA</td>
<td>7F6C8000</td>
<td>7FFFFFFF</td>
<td>9,440K</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Uncommitted Ext-Pvt</td>
<td>11A8A000</td>
<td>7F6C7FFF</td>
<td>1,798,392K</td>
<td>---</td>
<td>139FFFFF</td>
</tr>
<tr>
<td>AMTMS31 Committed Ext-Pvt</td>
<td>11A00000</td>
<td>11A89FFF</td>
<td>552K</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext CSA</td>
<td>051D0000</td>
<td>119FFFFF</td>
<td>204,992K</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext MLPA</td>
<td>0518A000</td>
<td>051CFFFF</td>
<td>88K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext FLPA</td>
<td>0271A000</td>
<td>02719FFF</td>
<td>14,856K</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext CSA</td>
<td>01898000</td>
<td>02719FFF</td>
<td>14,856K</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext NUC R/W</td>
<td>01596000</td>
<td>01597FFF</td>
<td>3,080K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Ext NUC R/O</td>
<td>00FDB000</td>
<td>00FFFFFF</td>
<td>148K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 SQA</td>
<td>00FC9000</td>
<td>00FDAFFF</td>
<td>72K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 MRA</td>
<td>00EBA000</td>
<td>00F8FFF</td>
<td>1,084K</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 MLP</td>
<td>00C0D000</td>
<td>00EC9FFF</td>
<td>2,740K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 CSA</td>
<td>00E8A000</td>
<td>00C0CFFF</td>
<td>524K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 LSQA/SWA</td>
<td>00B8A000</td>
<td>00B89FFF</td>
<td>3,624K</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Uncommitted Private</td>
<td>00800000</td>
<td>00889FFF</td>
<td>3,624K</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 Committed Private</td>
<td>00005000</td>
<td>0005FFF</td>
<td>360K</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 V-R (IF ANY)</td>
<td>00005000</td>
<td>00024FFF</td>
<td>128K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS31 System</td>
<td>00005000</td>
<td>00024FFF</td>
<td>128K</td>
<td>---</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. name of the address space
2. name of the virtual storage area
3. lowest virtual storage address for this storage area
4. highest virtual storage address for this storage area
5. number of kilobytes contained in this storage area
6. percent of the area in use, where available
7. region limit value
The MTP service displays volume serial numbers, device numbers, device types, and address space names for tape and direct access storage devices that have pending mount requests.

It also

- identifies the source of the malfunction when an address space stops processing
- finds out which address space requests a particular volume

**Syntax**

![MTP command]

**Example**

To identify pending mount requests, type

```
MTP
```

```
AMTM22I MOUNT PENDING FOR B90078 UNIT=384 (3400-4) JOB= GJJ$RUN
```

**Legend:**

1. volume serial number of the requested volume
2. number of the device that requires the volume mount
3. device type
4. name of the address space requesting the unmounted volume

**Note**

Only the volume serial number requested by the allocation appears in message AMTM22I. The second and successive volumes of a data set appear as blanks in message AMTM22I.

The NTP service displays Name/Token pair information for system, address space, and task-level Name/Token pairs.
## Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>specify SYSTEM, or no parameter, to display all system-level Name/Token pairs</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>seq#</td>
<td>previously displayed sequence number</td>
</tr>
</tbody>
</table>

When you specify a numeric value, the NTP service displays detailed information for the previously displayed Name/Token pair with that sequence number. See the examples for additional information.

### Example 1

To list all of the Name/Token pairs for an address space, enter the address space instance as the first parameter; for example:

**NTP,BMVDWP3**

<table>
<thead>
<tr>
<th>AMNTNT1</th>
<th>Token names for BMVDWP3 (ASID 013F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN21</td>
<td>Seq # Name(Character) Name(Hexadecimal)</td>
</tr>
<tr>
<td>AMNTN31</td>
<td></td>
</tr>
<tr>
<td>AMNTN41</td>
<td>1 *ISPFSCVX C9E2D7C6 E2E5C3E7 000004FC 00000001</td>
</tr>
<tr>
<td>AMNTN41</td>
<td>2 *XDCJZ1QXDC-XNTT E7C4C3BD E9F1C4D8 E7C4C360 E7D5E3E3</td>
</tr>
<tr>
<td>AMNTN41</td>
<td>3 *IEWBXL0/MMIB C9C5E6C2 E7C9D036 6D404AC9 2404040</td>
</tr>
<tr>
<td>AMNTN41</td>
<td>4 *SYSPROG 3.7 E2E8E2D7 D9D6C740 F34BF740 00000000</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN11</td>
<td>Token names for BMVDWP3 (013F) TCB 6FF070</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN21</td>
<td>Seq # Name(Character) Name(Hexadecimal)</td>
</tr>
<tr>
<td>AMNTN31</td>
<td></td>
</tr>
<tr>
<td>AMNTN41</td>
<td>5 *IRXTOKEN ?0 C9D9E7E3 D6D2C5D5 006FF070 2F309C90</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN11</td>
<td>Token names for BMVDWP3 (013F) TCB 6DE1E0</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN21</td>
<td>Seq # Name(Character) Name(Hexadecimal)</td>
</tr>
<tr>
<td>AMNTN31</td>
<td></td>
</tr>
<tr>
<td>AMNTN41</td>
<td>6 *IRXTOKEN _ C909E7E3 D6D2C5D5 006DE1E0 2F3AE9C90</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN11</td>
<td>Token names for BMVDWP3 (013F) TCB 6A33C0</td>
</tr>
<tr>
<td>AMNTN01</td>
<td></td>
</tr>
<tr>
<td>AMNTN21</td>
<td>Seq # Name(Character) Name(Hexadecimal)</td>
</tr>
<tr>
<td>AMNTN31</td>
<td></td>
</tr>
<tr>
<td>AMNTN41</td>
<td>7 *SYSPROG 3.5 NTP1 E2E8E2D7 D9D6C740 F34BF540 D5E3D7F1</td>
</tr>
<tr>
<td>AMNTN41</td>
<td>8 *SYSPROG 3.5 NTP2 E2E8E2D7 D9D6C740 F34BF540 D5E3D7F2</td>
</tr>
</tbody>
</table>

170  MainView SYSPROG Services User Guide and Reference
The names of the address-space level Name/Token pairs are displayed first, followed by the task-level token names for each task with associated Name/Token pairs. Use the sequence number to display the value of a token for a specific token name.

**Example 2**

To display all of the information, including the contents of the Name/Token pair identified with the sequence number 3 in the previous example, type

`NTP,3`

<table>
<thead>
<tr>
<th>AMNTNT51</th>
<th>Level=AdrsSpace BMVDP3 (ASID 013F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMNTNT61</td>
<td>Name =IEWXIL0/MMIB * 9C9E66C2 E7C9D006 61D4D0C9 24004040</td>
</tr>
<tr>
<td>AMNTNT71</td>
<td>Token =* { * 7F58FFC0 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

**Note**

The Name and Token values are displayed in both character and hexadecimal formats.

**Example 3**

To display all of the system-level token names, enter NTP without any parameters; for example:

`NTP`

<table>
<thead>
<tr>
<th>AMNTNT11</th>
<th>Token names for SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMNTNT01</td>
<td></td>
</tr>
<tr>
<td>AMNTNT02</td>
<td>Seq #  Name(Character) Name(Hexadecimal)</td>
</tr>
<tr>
<td>AMNTNT03</td>
<td></td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>1 <em>SYSZCUNDATASPACE</em> E2EB2E2E9 C3E4D5C4 C1E8C1E2 D7C1C3C5</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>2 *DSNLOGREC * C4E2D5D3 D6C7D9C5 C3A04040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>3 *AXMCSYS * C1E7D4E2 C3E8BEE2 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>4 *DSNDBGG * C4E2D5C4 C2E7C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>5 *DSNDBGH * C4E2D5C4 C2E7C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>6 *DSNDBGJ * C4E2D5C4 C2E7C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>7 *DSNDBGK * C4E2D5C4 C2E7C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>8 *DSNDHOG * C4E2D5C4 C8E6C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>9 *DSNDHH * C4E2D5C4 C8B8C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>10 *DSNDHW * C4E2D5C4 C6E6C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>11 *DSNDBI * C4E2D5C4 C5E2C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>12 *DSNID * C4E2D5C4 C5E2C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>13 *DSNIF * C4E2D5C4 C5E2C040 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>14 *DSNDED * C4E2D5C4 C5C4C440 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>15 *DSNDEEL * C4E2D5C4 C5C5D540 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>16 *DSNDEEN * C4E2D5C4 C5C5D540 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>17 *DSNDEOS * C4E2D5C4 C5C5D540 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>18 *DFSMSTPC.ManagRMT * C4C6E2D4 E2E3D7C3 4BD48195 8167D9D4</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>19 *IBMJESXCFAS * C9C2D4D1 C5E2E7C3 C6C1E240 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>20 *AXRWORD * C1E7D9E6 D6D9D3C4 40404040 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>21 *JES2_AUXECB_JES2 * D1C5E2F2 6DC1E4E7 C5C3C26D D1C5E2F2</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>22 *JES2_LX_NUM_JES2 * D1C5E2F2 6DC1E4E7 D5E4D46D D1C5E2F2</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>23 *ATRRRS.STOKEN * C1E3D9D9 D9E4BE2E E3D6D2C5 D3D4D040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>24 *ISFHSVT.SDSF * C9E2C6C8 E2E5E3D4 E2C4E2C6 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>25 *ISFHLX.SDSF * C9E2C6C8 D3E74BE2 C4E2E640 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>26 *ISFHMS.SDSF * C9E2C6C8 E7D4E24B E2C4E2C6 40404040</td>
</tr>
<tr>
<td>AMNTNT41</td>
<td>27 *ISFQSRV.SDSF * C9E2C6C8 E2D9E54B E2C4E2C6 40404040</td>
</tr>
</tbody>
</table>
OFIND

The OFIND (OFI) service searches a memory object for a specified string.

You can specify an end address as well as a beginning address. If you do not specify an end address, the search continues to the end of the memory object. For the search, you can also specify a boundary alignment and an offset from that alignment.

**Note**

Before using OFIND, you must use the OLIST service to create symbolic names for the memory objects in the subject address space. See OLIST on page 176 for details.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StrExp</strong></td>
<td>can be either an expression that denotes the location and length of the subject search string or a literal expressed as a character or hexadecimal value. Character values are converted to uppercase unless enclosed in quotation marks. You must enclose character strings that contain special characters (for example, blanks, commas, and parentheses) in quotation marks. When enclosed in quotation marks, a &quot;C&quot; can optionally precede character strings. Hexadecimal values must be enclosed in single quotation marks and preceded by an X (for example: X'0A0B'). Either an expression or a literal value is required for an initial search. You can specify an asterisk when resuming a search. The search string can be from 1 to 64 bytes in length. By using expressions, you can refer to the search string if it is already in storage, thereby avoiding the need to type in the string itself. <strong>Note:</strong> The search string must be either in common storage or in the private area of the address space that is implied by the second parameter. It can be either below or above the bar. Expressions are especially useful for large, hexadecimal values that can be up to 128 characters (64 bytes) in length. Expressions are differentiated from a character literal by the presence of the string length enclosed within parentheses at the end of the expression. For example, ASCB+4(4) indicates that the search string is four bytes in length and starts at ASCB+4. Also see Example 3 on page 174.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
---|---
**Begin** | required for the initial search; defines the starting location for the search and can be any expression that resolves within a memory object; for example: MOBJ0002 or MOBJ0002+1M
| Omit the parameter or use * to resume the previous search.

**End** | *(optional)* specifies the ending address for the search; can be an expression that resolves to a location within the memory object defined by the *begin* expression
| a value that when added to the beginning of the memory object defined by the *begin* expression, falls within the bounds of that memory object
| If omitted, the end-of-memory object is assumed.

**BNDRY** | *(optional)* specifies the boundary alignment (*m*) and offset from that alignment (*n*) for the search
| An *m* can be 1 or any multiple of 2, for example: 1, 2, 4, 8, 16, 32, and 64. Larger alignments can also be expressed by using the scalars K and M. For example: 1K for 1024, 4K for 4096, and 1M for 1,048,576. 1M is the largest valid boundary alignment.
| The offset value, *n*, is optional.
| If specified, it must be a value from 0 to *m*-1. The default for *m* is 1, which indicates single-byte alignment. The default value for *n* is 0.
| **BNDRY(8)** is equivalent to **BNDRY(8,0)** which is at the beginning of each double-word. **BNDRY(8,2)** means displaced two bytes from each double-word boundary.

---

**Example 1**

Find the first occurrence of Obj within the first five megabytes of memory object MOBJ0001.

**Note**

Because the search string Obj is enclosed by single quotation marks, it is *not* converted to uppercase before initiating the search.

**OFIND,'Obj',MOBJ0001,5M**

*Found at: 00000001_00000008= MOBJ0001+8*

*Storage Obj1 - P*

*INION - 10/10/03*
Example 2

You suspect that memory object MOBJ0005 contains control blocks that are identified by the acronym WPHG. You also know that the control blocks begin on a page boundary and that the acronym is displaced four bytes from the beginning of the control block. You want to find the first occurrence of this block in the memory object, so you type

**OF,WPHG.MOBJ0005,BNDRY(4K,4)**

You notice that there are double-word address pointers at offsets +8 and +10 that point to locations within the memory object; you decide to examine the contents of the storage referenced by the second pointer. You know that the last-referenced address variable (*) is currently pointing to the beginning of the block that you just found, so you can use the expression *+10 to reference the second pointer. Because you want to display the storage at that 64-bit address, you add the 64-bit indirection indicator (!) to the end of that expression and use that expression as the first parameter of the DUMP service, as follows:

**DUMP *+10!**

Example 3

Suppose that you want to search the memory object for other occurrences of the eight-byte address pointer located at +10 in MOBJ0005 in the previous example. You do not want to type in the hexadecimal value X'0000000541C02000', so you elect to reference it by using an expression. Furthermore, you are only interested in finding occurrences that are on a double-word boundary. You want to search beginning immediately after the string that you are referencing, and search through the entire memory object, if necessary. So, you type

**OFIND,MOBJ0005+10(8), MOBJ0005+18, BNDRY(8)**

**OKSWAP**

The OKSWAP (OKS) service reverses the effects of the DONTSWAP service, making an address space swappable.
Syntax

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

Note
If you do not specify an address space to be made swappable, your address space is changed.

Examples

To reduce the nonswappability count of address space INVENTRY, type OKSWAP, INVENTRY

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS61I TSU06701</td>
<td>INVENTRY</td>
<td>IKJDWP1</td>
<td>Service Class=TSO/NRM</td>
</tr>
<tr>
<td>AMTS64I BMVDWP4</td>
<td>NONSWAPPABILITY COUNT REDUCED (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT001A SYSPROG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To make address space INVENTRY swappable, type OKSWAP, INVENTRY

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS61I TSU06701</td>
<td>INVENTRY</td>
<td>IKJDWP1</td>
<td>Service Class=TSO/NRM</td>
</tr>
<tr>
<td>AMTS62I BMVDWP4</td>
<td>HAS BEEN SET SWAPPABLE (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT001A SYSPROG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note
The address space is swappable when the non-swappability count is zero (as shown on line AMTS62I in the preceding example).

Legend:

1 JES job ID, address space name, and current stepname of the specified address space

2 current service class for the address space

3 nonswappability count

The operating system keeps this count for the address space. If the count is greater than zero, the address space is still nonswappable. The OKSWAP service reduces the count by one. When the count reaches zero, the address space is swappable.
Usage notes

- The OKSWAP service reverses the effects of the DONTSWAP service.
- Using the OKSWAP service to make system-specified nonswappable address spaces swappable is not recommended.

OLIST

The OLIST (OLI) service displays information about the Large Virtual Memory Objects within an address space and shared memory objects that are associated with the address space.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>specifies your own address space</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>
| symbol    | symbolic name assigned by the OLIST service to all memory objects owned by an address space  
The symbolic names begin with MOBJ, followed by a four-digit decimal. You can reference the memory object by its symbolic name and use the symbol in expressions. |
| ALL       | all memory objects owned by all address spaces |
| DATE      | (optional) time and date that the memory objects were created (default) |
| TOKEN     | (optional) causes the user tokens associated with the memory objects to be displayed instead of the time and date |
Examples

To display a list of the memory objects in or associated with address space X00AB and the time and date that the objects were created, type

OLIST,(X00AB)

To display a list of the memory objects in job name MVQAZPAS and the user token for each object, type

OLIST,MVQAZPAS,TOKEN

To display detailed information for a specific memory object, type

OL,MOBJ0002

If you specify the symbol name for the memory object, the OLIST service will display all of the available information for that memory object.

To display all of the memory objects owned by all address spaces, type

OLIST,ALL
Legend:

1 a sequential number assigned by the OLIST service
   The OLIST service creates a symbolic definition for each memory object, in the
   form MOBJ nnnn. Symbols can be used with the DUMP, EQUATE, and ZAP
   services. For more details, see “Symbolic references” on page 179.

   Note
   The DUMP service can also generate symbol names in the form MOBJ nnnn.

2 the lowest accessible virtual address for the memory object
   If the memory object contains guard segments at the beginning of the object
   (GUARDLOC=LOW), then ORIGIN is the address of the first segment after the
   guard segments.

3 the total size of the accessible segments (total segments minus guard segments)
   within the memory object

4 the size of the guard segments, if any
   The size is followed by a less than sign (<) or a greater than sign (>) to denote the
   location of the guard segments (low or high end of the object).

5 the storage key for the memory object

6 for private memory objects, the TCB address for the task that owns the memory
   object
   The memory object will be detached by the Recovery/Termination Manager
   automatically when the task terminates. The word Shared indicates that the
   memory object is a shared memory object.

7 time and date that the memory object was created
   The format is hh:mm for the time and mm/dd/yyyy for the date.

8 the user token associated with the memory object when it was created

9 the guard location (high or low)
   This value is present even if there are no guard segments.

10 the job name and Started Task ID (jobName.stid)
   This value is displayed when the address space contains a Started Task and the
   Started Task ID is other than the job name; in all other cases, only the job name is
   displayed.

11 the address space identifier (ASID) expressed in hexadecimal
Symbolic references

The OLIST service generates a symbolic reference for each memory object. You can display a list of the generated names and the associated storage addresses by executing the DUMP service without any operands. Here is an example:

DUMP

<table>
<thead>
<tr>
<th>Symbolic reference</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1GI MOBJ0001</td>
<td>000000001_00100000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0002</td>
<td>000000001_00400000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0003</td>
<td>000000001_01500000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0004</td>
<td>000000001_01E00000</td>
</tr>
</tbody>
</table>

You can subsequently use these symbolic names to display the storage; for example:

DU,MOBJ0001+60!

You can subsequently use these symbolic names to display the storage; for example:

PAGING

The PAGING (PAG) service is the paging monitor for the z/OS system.

This service provides the total paging rate and divides the total paging rate into its components for analysis. It also provides the page reclaim rate.

Syntax

Example

To display paging information, type

PAGING

<table>
<thead>
<tr>
<th>Symbolic reference</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTP111 Total paging rate</td>
<td>[1] 2.20 Pages/Sec</td>
</tr>
<tr>
<td>AMTP141 Local paging rate on SCM</td>
<td>[2] 2.18 Pages/Sec</td>
</tr>
<tr>
<td>AMTP121 Local paging rate on DASD</td>
<td>0.35 Pages/Sec</td>
</tr>
<tr>
<td>AMTP131 VI0 paging rate</td>
<td>0.00 Pages/Sec</td>
</tr>
<tr>
<td>AMTP141 Swap paging rate</td>
<td>0.00 Pages/Sec</td>
</tr>
<tr>
<td>AMTP151 LPA paging rate</td>
<td>0.00 Pages/Sec</td>
</tr>
<tr>
<td>AMTP161 Common paging rate</td>
<td>0.00 Pages/Sec</td>
</tr>
<tr>
<td>AMTP171 Page reclaim rate</td>
<td>0.00 Pages/Sec</td>
</tr>
</tbody>
</table>
Legend:

1. total system paging rate over the sample interval
2. length of the sample interval
3. local Paging rate on Storage Class Memory (SCM). This message is only displayed when SCM is present.
4. local paging rate on traditional direct access storage devices (DASD)
5. virtual I/O (VIO) paging rate
6. swap paging rate
7. Link Pack Area (LPA) paging rate.
8. Common Storage Area (CSA) paging rate
9. page reclaim rate

Usage note

The period for which paging data is reported varies, based on the time that the paging activity counters were last cleared.

PARMLIST

The PARMLIST (PAR) service lists the contents of SYS1.PARMLIB members.

This service helps you examine system initialization and operation parameters.

Syntax

```
PARMLIST member
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>member in SYS1.PARMLIB</td>
</tr>
</tbody>
</table>
Example

To list the contents of SYS1.PARMLIB member IEASYS00, type

```
PARMLIST IEASYS00
```

```
AMTCFI1 APF=00,
AMTCFI1 CMD=00,
AMTCFI1 CSA=3072,
AMTCFI1 CVIO,
AMTCFI1 FIX=00,
AMTCFI1 ICS=00,
AMTCFI1 IOS=00,
AMTCFI1 IPS=00,
AMTCFI1 LNK=00,
AMTCFI1 MLPA=(00),
AMTCFI1 LOGCLS=L,
AMTCFI1 LOGLMT=050000,
AMTCFI1 MAXUSR=50,
AMTCFI1 OPT=00,
AMTCFI1 PAGE=(SYS1.PAGEA,
AMTCFI1     SYS1.LOCALA,
AMTCFI1     SYS1.LOCALB,
AMTCFI1     SYS1.LOCALC,
AMTCFI1     SYS1.LOCALD),
AMTCFI1 PAGNUM=(4,3),
AMTCFI1 REAL=192,
AMTCFI1 RSU=2,
AMTCFI1 SMF=00,
AMTCFI1 SQA=8,
AMTCFI1 SWAP=(SYS1.SWAPA
AMTCFI1     SYS1.SWAPB),
AMTCFI1 VAL=00,
AMTCFI1 VRREGN=192,
```

PBUILD

The PBUILD (PLB) service provides the ability to create or modify exclusion lists after the initialization of SYSPROG Services by processing the contents of an exclusion list member in a partitioned data set.

The supported exclusion lists are $$INEXCL, $$PVT, $$RPJOBS, $$RPTEXT, $$XDEV, $$XENQ, and $$XRES.

Syntax

```
PBUILD member dsn
```

The parameters are defined as follows:
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>the member to be processed ($$$INEXCL, $$PVT, $$RPJOBS, $$RPTEXT, $$XENQ, $$XDEV, or $$XRES)</td>
</tr>
<tr>
<td>dsn</td>
<td>(optional) name of the partitioned data set that you want to use. Do not enclose the name in quotes. Provide this name if a parameter library was not defined prior to starting SYSSPROG services or if you prefer another library. If you omit dsn, PBUILD uses the parameter library that was in existence when SYSSPROG Services was initialized.</td>
</tr>
</tbody>
</table>

**Note**

You can use the PLIST service to display the contents of an active exclusion list.

### Example

To build an exclusion list from the contents of member $$$INEXCL, type

```
PBUILD, $$$INEXCL
```

AMTPB11 $INEXCL successfully processed

---

**PFKEY**

The PFKEY (PFK) service displays or changes the definitions of the program function (PF) keys in effect when you are using MainView SYSPROG Services in TSO mode.

This service does *not* function in the MainView environment.

### Syntax

```
PFKEY LIST nn command
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>displays all of the current PF key definitions</td>
</tr>
<tr>
<td>nn</td>
<td>indicates the number of the PF key whose definition you want to change</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>command</td>
<td>indicates the SYSPROG command to be invoked by the particular program function key</td>
</tr>
</tbody>
</table>

**Example**

To display the current PF key definitions, type

**PFK LIST**

```
AMTP81I TR
AMTP81I PA
AMTP81I 01 ENQUEUES
AMTP81I 02 RESERVES
AMTP81I 03 END
AMTP81I 04 CSA,MAP
AMTP81I 05 MTP
AMTP81I 06 PAGING
AMTP81I 07 STATUS
AMTP81I 08 REPLIES
AMTP81I 09 SRM
AMTP81I 10 CPU,5
AMTP81I 11 CMD,||
AMTP81I 12 IO
AMTP81I 13 ENQUEUES
AMTP81I 14 RESERVES
AMTP81I 15 END
AMTP81I 16 CSA,MAP
AMTP81I 17 MTP
AMTP81I 18 PAGING
AMTP81I 19 STATUS
AMTP81I 20 REPLIES
AMTP81I 21 SRM
AMTP81I 22 CPU,5
AMTP81I 23 CMD,||
AMTP81I 24 IO
```

To change the definition of PF key 01 so that it invokes the STATUS service with ALL as an operand, type

**PFK 01 ‘STATUS ALL’**

```
AMTP82I PFKEY DEFINITION CHANGED FOR NO. 1
```

**Note**

Commands containing commas or embedded blanks must be enclosed in quotation marks.

---

**PIO**

The PIO service monitors I/O interrupt activity and determines the distribution of I/O interrupts among processors.
Syntax

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 999 seconds</td>
</tr>
</tbody>
</table>

Example

To display I/O interrupt activity, type

PIO

AMTO70I I/O Interrupt data being gathered for 15 seconds

AMTO73I Processor I/O Interrupt Activity Report
[1] [2] [3]
AMTO74I CPU 00 processed 676 interrupts (  0% were through TPI)
AMTO74I CPU 10 processed 1,116 interrupts ( 1% were through TPI)

Legend:

1 CPU identifier (CPUID) for the processor whose activity is being reported

2 total number of I/O interrupts (including TPIs) processed by this CPU in the interval

3 percentage of I/O interrupts processed by this CPU through the Test Pending Interrupt (TPI) instruction—the percentage of I/O interrupts that do not require a physical program status word (PSW) swap.

PLIST

The PLIST (PLI) service displays the active exclusion list for $$INEXCL, $$PVT, $$RPJOBS, $$RPTEXT, $$XDEV, $$XENQ, and $$XRES.

The service also displays the name of the list’s creator and the time and date when the list was created.
Syntax

```
PLIST member
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>the member to be processed ($$INEXCL, $PVT, $$RPJOBS, $$RPTEXT, $$XDEV, $$XENQ, or $$XRES)</td>
</tr>
</tbody>
</table>

Example

To display the active exclusion list $$INEXCL, type

```
PLIST, $$INEXCL
```

AMTPL2I Exclusion list $$INEXCL created by BMVDWP3
AMTPL2I Monday, May 21, 2012; 1:35:03.69 PM Local
AMTPL2I Sampler Excluded Jobnames
AMTPL2I AMTPL3I AI0 ---JES2
AMTPL2I AMTPL3I AIOR---JES2
AMTPL2I AMTPL3I APG ---MTACDMD7 MTACDMD8 MTA$DBCD

POST

The POST service allows you to post any desired ECB in any address space.

**Note**

The TCB service locates and assigns symbolic names to the ECBs upon which a task is waiting when it displays detail information for a specific task. See Example 2 on page 253 in the description of the TCB service.

Syntax

```
POST expression si
```

The parameters are defined as follows:
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| expression | addressing value that resolves to an ECB address  
It can and usually is a symbol created by the TCB service. The TCB service locates and assigns symbolic names to ECBs that are in a waiting state. For more information about the TCB service, see “TCB” on page 252.  
The combination of expression and asi must resolve to a specific ECB, in a specific task, in a specific address space. Otherwise, the post fails. |
| asi | address space instance  
**Best practice**: BMC recommends using address space tokens the JITOKEN service locates and assigns because such tokens are unique. For more information, see “JITOKEN” on page 131. For example, if you specify an address space that has ended and been reused, the post fails.  
In general, you do not need to specify this parameter if you use the TCB service to locate the task and waiting TCB, because the TCB service defines the address space instance when it creates the symbol. |
| code | *(optional)* post code  
It can be a decimal or hexadecimal value.  
If omitted, this parameter defaults to zero (0). |

### Example

Assume that the following AMTT76I messages were displayed for a TCB in the MYSPACE address space:

AMTT76I The task is Waiting on ECB1 at 006C1D28  
AMTT76I The task is Waiting on ECB2 at 00050ED4  
AMTT76I The task is Waiting on ECB3 at 006C1D50  

As a result of the information in these messages, both of the following commands will post the ECB defined by the first AMTT76I message using the post code of 16:

POST,ECB1,16  
POST,006C1D28,MYSPACE,16

### PPT

The PPT service displays and updates the program properties table (PPT).

The PPT service is used to

- display the entire program properties table (PPT)
- display the PPT entry for all programs that begin with a string of one or more characters
display the PPT entry for a specific program

functionally delete a PPT entry (remove all special attributes)

add or change PPT entries; the presence of one or more keywords in addition to the program name indicates the following situations:

— A new entry will be created, if one does not currently exist.

— If an entry exists, it should be updated according to the keywords that were specified. Only the specified attributes are changed.

When a new entry is created, all attributes are set to the IBM default value prior to applying the specified keywords. Keywords that begin with NO can be abbreviated to their first five characters. Other keywords can be abbreviated to their first three characters. If neither keyword in a row is specified, the existing value is retained.

Syntax

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL</td>
<td>CANCEL is the IBM default value; NOCANCEL is the opposite value.</td>
</tr>
<tr>
<td>NOCANCEL</td>
<td></td>
</tr>
<tr>
<td>DSI</td>
<td>DSI is the IBM default value; NODSI is the opposite value.</td>
</tr>
<tr>
<td>NODSI</td>
<td></td>
</tr>
<tr>
<td>HUSI</td>
<td>HUSI is the IBM default value that honors IEFUSI exits. NOHUSI is the opposite value that does not honor IEFUSI exits.</td>
</tr>
<tr>
<td>NOHUSI</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>NOLPREF</td>
<td>NOLPREF or NO1P is the IBM default value; LPREF or 1P is the opposite value.</td>
</tr>
<tr>
<td>LPREF</td>
<td></td>
</tr>
<tr>
<td>NOPRIV</td>
<td>NOPRIV is the IBM default value; PRIV is the opposite value.</td>
</tr>
<tr>
<td>PRIV</td>
<td></td>
</tr>
<tr>
<td>NOSPREF</td>
<td>NOSPREF or NO2P is the IBM default value; SPREF or 2P is the opposite value.</td>
</tr>
<tr>
<td>SPREF</td>
<td></td>
</tr>
<tr>
<td>NOSYSTAS</td>
<td>NOSYSTAS is the IBM default value; SYSTASK is the opposite value.</td>
</tr>
<tr>
<td>SYSTASK</td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>PASS is the IBM default value; NOPASS is the opposite value.</td>
</tr>
<tr>
<td>NOPASS</td>
<td></td>
</tr>
<tr>
<td>PREF</td>
<td>PREF is the IBM default value; NOPREF is the opposite value.</td>
</tr>
<tr>
<td>NOPREF</td>
<td></td>
</tr>
<tr>
<td>SWAP</td>
<td>SWAP is the IBM default value; NOSWAP is the opposite value.</td>
</tr>
<tr>
<td>NOSWAP</td>
<td></td>
</tr>
<tr>
<td>KEY(nn)</td>
<td>Value of nn is 0-15. (The default is 8.)</td>
</tr>
<tr>
<td>RESET</td>
<td>RESET indicates that all IBM defaults are to be applied to the entry (all special attributes removed). RESET cannot be specified in conjunction with any other keyword.</td>
</tr>
</tbody>
</table>

The following table shows how you can use the PPT service.

<table>
<thead>
<tr>
<th>Type</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>display all entries in the PPT</td>
</tr>
<tr>
<td>PPT, prefix*</td>
<td>display all PPT entries that begin with the prefix</td>
</tr>
<tr>
<td>PPT, name</td>
<td>display the PPT entry for a program name</td>
</tr>
<tr>
<td>PPT, name,RESET</td>
<td>reset all of the attributes in the PPT entry for the program name to the IBM defaults, which effectively deletes the entry</td>
</tr>
<tr>
<td>PPT, name,attrl,attr2,...attrn</td>
<td>create a PPT entry for the program name, if one does not already exist, and set the specified attributes; unspecified attributes remain as currently defined. For example: PPT, name, NOSWAP,CANCEL would add the NOSWAP attribute, remove the NOCANCEL attribute, and leave all others unchanged.</td>
</tr>
<tr>
<td>PPT, name,NOHUSI</td>
<td>modify the PPT entry for the program name to ignore IEFUSI region controls</td>
</tr>
</tbody>
</table>
Example 1

To display all the PPT entries, type

```
PPT
```

<table>
<thead>
<tr>
<th>AMTPP1I Program</th>
<th>Non</th>
<th>Non</th>
<th>SYS</th>
<th>DS</th>
<th>Bypass</th>
<th>HUSI</th>
<th>Storage</th>
<th>Pref</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPP2I Name</td>
<td>Cancel</td>
<td>Swap</td>
<td>Key</td>
<td>Priv</td>
<td>Task</td>
<td>Integ</td>
<td>Pswd</td>
<td></td>
</tr>
<tr>
<td>AMTPP3I --------</td>
<td>------</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>------</td>
<td>-----</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>AMTPP4I IEDQTCAM</td>
<td>No</td>
<td>Yes</td>
<td>6</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>AMTPP4I IGETZ17C</td>
<td>Yes</td>
<td>Yes</td>
<td>6</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AMTPP4I HHLGTF</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AMTPP4I IHLGTF</td>
<td>Yes</td>
<td>No</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AMTPP4I IEAVTDSV</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AMTPP4I IEFIC</td>
<td>Yes</td>
<td>No</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Example 2

To display all PPT entries beginning with IEDQ, type

```
PPT,iedq*
```

Example 3

To reset all attributes for program name DWPTEST, if it exists in the table, type

```
PPT,DWPTEST,RESET
```

AMTPP9I Add/Update successful.
Note
If DWPTTEST does not currently exist, it is created.

Example 4

To set two attributes in the entry for DWPTTEST, if it exists in the table, type

**PPT,DWPTTEST,NOCAN,NOSWAP**

```
AMTPP9I  Add/Update successful.
AMTPP0I
AMTPP1I Program  Non  Non                SYS  DS  Bypass  HUSI  Storage  Pref
AMTPP2I  Name    Cancel  Swap  Key  Priv  Task  Integ  Pswd
AMTPP3I   --------  ------  ----  ---  ----  ----  -----  ------  ----  ------------
AMTPP4I  DWPTTEST  Yes  Yes  8  No  No  Yes  No  Yes
AMTPP0I
AMT001A SYSPROG
```

Note
If DWPTTEST does not currently exist, it is created.

Example 5

To change the key for entry DWPTTEST, if it exists, type

**PPT,DWPTTEST,KEY(4)**

```
AMTPP9I  Add/Update successful.
AMTPP0I
AMTPP1I Program  Non  Non                SYS  DS  Bypass  HUSI  Storage  Pref
AMTPP2I  Name    Cancel  Swap  Key  Priv  Task  Integ  Pswd
AMTPP3I   --------  ------  ----  ---  ----  ----  -----  ------  ----  ------------
AMTPP4I  DWPTTEST  Yes  Yes  4  No  No  Yes  No  Yes
AMTPP0I
AMT001A SYSPROG
```

Note
If DWPTTEST does not currently exist, it is created.

PRIVATE

The PRIVATE (PRI) service summarizes virtual storage allocations for private storage in an address space and displays an overview of subpool allocations within a private address space.

Use the PRIVATE service to

- display allocated storage and fragmented free space by subpool and key
- debug applications that use excessive private storage
- tune virtual storage-constrained systems
- monitor virtual storage use within an address space

## Syntax

```
PRIVATE  @  asi  \rightarrow  LSQA  \rightarrow  spid  TCB  \rightarrow  address
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>your own address space (default)</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>LSQA</td>
<td>detailed report of allocations in the Local Systems Queue Area (LSQA), which consists of subpools 203-205, 213-215, 223-225, and 253-255</td>
</tr>
<tr>
<td>spid</td>
<td>three-digit decimal subpool identifier in the range between 0 and 255</td>
</tr>
<tr>
<td>TCB</td>
<td>summary of the storage allocated by all tasks in the address space</td>
</tr>
<tr>
<td>address</td>
<td>address of the TCB for which you want to display a detailed report of storage allocations</td>
</tr>
<tr>
<td>HIGH</td>
<td>summary of subpools located in private high storage</td>
</tr>
<tr>
<td>LOW</td>
<td>summary of subpools located in private low storage</td>
</tr>
<tr>
<td>PVT</td>
<td>summary of subpools 0-127, 129-132, 229, 230, 236, 237, 251, and 252</td>
</tr>
<tr>
<td>SWA</td>
<td>detailed report of allocations in the Scheduler Work Area (SWA), which consists of subpools 236 and 237</td>
</tr>
<tr>
<td>REGION</td>
<td>regions for the specified address space</td>
</tr>
</tbody>
</table>

### Example 1

To determine LSQA usage for address space INVENTRY, type

```
PRIVATE INVENTRY
```

<table>
<thead>
<tr>
<th>SP</th>
<th>TYPE</th>
<th>KEY</th>
<th>ABOVE</th>
<th>BELOW</th>
<th>OWNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TYPE</td>
<td>KEY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TCB</td>
<td>INVENTORY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RB</td>
<td>7FE158,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PGM</td>
<td>IEAVAR00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ALLOC</td>
<td>FE158,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>FFS</td>
<td>RB=007FDC58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PGM</td>
<td>IEFSD060</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ALLOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ALLOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example 2**

To summarize storage allocations by subpool and key for all TCBs belonging to address space INVENTORY, type

**PRIVATE INVENTORY TCB**

<table>
<thead>
<tr>
<th>SP</th>
<th>TYPE</th>
<th>KEY</th>
<th>ABOVE</th>
<th>BELOW</th>
<th>OWNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TYPE</td>
<td>KEY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TCB</td>
<td>INVENTORY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RB</td>
<td>7FE158,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PGM</td>
<td>IEAVAR00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ALLOC</td>
<td>FE158,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>FFS</td>
<td>RB=007FDC58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PGM</td>
<td>IEFSD060</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ALLOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ALLOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FFS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. type of private storage information displayed
2. address space name for which private storage information is displayed
3. amount of allocated storage, both above and below the 16-MB line, for a subpool within the key
4. amount of fragmented free space, both above and below the 16-MB line, within the allocated storage for a subpool within the key
5. total amount of allocated storage for the address space by subpool and key
6. total amount of fragmented free space within allocated storage for the address space by subpool and key

MainView SYSPROG Services User Guide and Reference
Legend:

1. address of TCB
2. address of current RB
3. name of program running under the current RB
4. subpool number
5. key
6. whether subpool is owned (OWN) or shared (SHR)
7. amount of storage allocated above the 16-MB line
8. amount of free space above the 16-MB line
9. amount of storage allocated below the 16-MB line
10. amount of free space below the 16-MB line
11. total of all allocated and free storage for the TCB, in all subpools and keys

Example 3

To display storage allocations by subpool and key for the specific TCB at location 7FE158, type

PRIVATE INVENTORY TCB 7FE158
Legend:

1 address to which the TCB belongs

2 subpool number

3 key

4 whether subpool is owned (OWN) or shared (SHR)

5 address of storage block allocated by the TCB in the subpool and key

6 length of storage block

7 first 32 bytes of storage block

8 character representation of storage block

**Example 4**

To list the regions for address space INVENTRY, type
PRIVATE INVENTORY REGION

<table>
<thead>
<tr>
<th>REGION</th>
<th>ADDRESS</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCT</td>
<td>00001000</td>
<td>00004000</td>
</tr>
<tr>
<td>V-V</td>
<td>00005000</td>
<td>007FB000</td>
</tr>
<tr>
<td>E V-V</td>
<td>05800000</td>
<td>7A800000</td>
</tr>
</tbody>
</table>

PROGRESS

The PROGRESS (PRO) service displays status and current step information for the specified address space. This service helps you determine how long an address space or step has been running.

Two versions of the PROGRESS service are supported:

- **Version 3**: the default when the PROGRESS service is invoked under MainView for z/OS, MainView SYSPROG Services, or as a TSO command processor
- **Version 2**: the default when the PROGRESS service is invoked through MainView AutoOPERATOR

You can use the version parameter to invoke a version different from the default.

**Syntax**

```
PROGRESS asi version
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>Address space instance; see “Identifying address spaces” on page 47 for more information.</td>
</tr>
<tr>
<td>version</td>
<td>version number</td>
</tr>
</tbody>
</table>

Specifying:

- **3** for the default version when the PROGRESS service is invoked under MainView for z/OS, MainView SYSPROG Services, or as a TSO command processor
- **2** for the default version when the PROGRESS service is invoked through MainView AutoOPERATOR

If you do not specify a `version`, the default version of the PROGRESS service is invoked.
Example 1 - version 3

In this example, the PROGRESS service is invoked under MainView for z/OS, MainView SYSPROG Services, or as a TSO command processor. To display the progress of address space BMVDWP4 (which happens to be a TSO session), type PROGRESS

**BMVDWP4**

<table>
<thead>
<tr>
<th>Message</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPG0I</td>
<td>TSO03895 BMVDWP4</td>
</tr>
<tr>
<td>AMTPG1I</td>
<td>Swaped out due to: Long wait</td>
</tr>
<tr>
<td>AMTPG3I</td>
<td>MSGCLASS= R MSGLEVEL=(1,1)</td>
</tr>
<tr>
<td>AMTPG7I</td>
<td>Program=IKJEFT01 Pgmr Name=</td>
</tr>
<tr>
<td>AMTPG11 Job Start Date=</td>
<td>24 May 2012 2012.145</td>
</tr>
<tr>
<td>AMTPG21 Job Start Time=</td>
<td>8:47:26.960 Job Class= *</td>
</tr>
<tr>
<td>AMTPG41 Step Start Time=</td>
<td>8:47:26.960 Step= IKJDWP2 (1/1)</td>
</tr>
<tr>
<td>AMTPG51 Step TCB Time=</td>
<td>0:00:00.716 Step SRB Time= 0:00:00.045</td>
</tr>
<tr>
<td>AMTPG61 Step TCB Limit=</td>
<td>23:59:00.796 Pct. used= 0.00%</td>
</tr>
<tr>
<td>AMTPG81 Step Phtm Time=</td>
<td>0:00:00.002 ASST on CP= 0:00:00.000</td>
</tr>
<tr>
<td>AMTPG91 Step ASST Time=</td>
<td>0:00:00.000</td>
</tr>
<tr>
<td>AMTPG61 Region Requested=</td>
<td>64Mi (Only applies to bottom of Rgn)</td>
</tr>
<tr>
<td>AMTPG51 Rgn Bot Limit Below=</td>
<td>7,144Ki Cur Bot Rgn Below=</td>
</tr>
<tr>
<td>AMTPG41 Max Bot Rgn Below=</td>
<td>456Ki Cur Bot Rgn Above=</td>
</tr>
<tr>
<td>AMTPG31 Cur Top Rgn Below=</td>
<td>504Ki Cur Top Rgn Above=</td>
</tr>
<tr>
<td>AMTPG71 Common Storage</td>
<td>---CSA---- ---SQA----</td>
</tr>
<tr>
<td>AMTPG91 Below</td>
<td>136 96</td>
</tr>
<tr>
<td>AMTPG71 Above</td>
<td>3,656 368</td>
</tr>
<tr>
<td>AMTPG8I LNKLST Set Name= IPL</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. job ID, job name, and service class name
2. whether the address space is reusable
3. message class and message level
4. name of the program attached by the current job step, and the programmer's name from the JOB statement
5. job start date in Gregorian and Julian formats
6. job start time and the job class, if applicable
7. start time of the current step, step name, and current step number divide by the total number of steps
8 amount of TCB time accumulated by the job step, followed by the amount of SRB time attributed to the job step (the time used by SRBs that are executing in the address space)

9 step TCB time limit. This is the maximum amount of TCB time the system will allow the step to accrue before it is terminated with a 322 ABEND. The second value is the percentage of the limit that has been used

10 amount of zAAP time used and the percentage of conventional CP time that was zAAP eligible

11 amount of zIIP time used and the percentage of conventional CP time that was zIIP eligible

12 preemptable-class SRB time

13 ASST = Additional SRB Service Time. The second field is the amount of ASST time on a CP rather than a specialty processor

14 Region size requested
This value applies only to the subpools that are allocated from the bottom of the region. This value does not include sub pools that are allocated from the top of the region (such as LCQA and SWA)

15 storage limits for "bottom of region" storage, both below-the-line and above-the-line

16 amount of storage currently allocated from the bottom of the region below-the-line and the bottom of the region above-the-line

17 high-water-mark (maximum) of "bottom of region" storage allocated for the address space below-the-line and above-the-line

18 amount of storage currently allocated from the top of the region below-the-line and the top of the region above-the-line

19 high-water mark (maximum) of "top of region" storage allocated for the address space below-the-line and above-the-line

20 header line that labels the columns for the subsequent two lines

21 amount of below-the-line CSA and SQA storage currently owned by (attributed to) the address space

22 amount of above-the-line CSA and SQA storage currently owned by (attributed to) the address space
maximum amount of above-the-bar private storage that the address space may allocate and how that limit was established

current amount of above-the-bar private storage allocated by the address space and the amount that is hidden

high-water mark (maximum amount) of above-the-bar-storage allocated by this address space

name of the address space’s LNKLST set

message that is displayed when the address space has been quiesced

message that indicates the swap reason displayed when the address space is swapped out

Example 2 - version 3

In this example, only a limited amount of information is displayed for the address space because in this instance the address space has been quiesced. When quiesced, the address space’s private storage is not accessible. Also, the PROGRESS service is invoked under MainView for z/OS, MainView SYSPROG Services, or as a TSO command processor.

PRO,BMVDWP4

For the legend that explains this output, see the legend for Example 1 - version 3 on page 196.

Example 3 - version 2

In this example, the PROGRESS service is invoked through MainView AutoOPERATOR. To display the progress of TSO user XTSTWPAS, type PRO,XTSTWPAS

For the legend that explains this output, see the legend for Example 1 - version 3 on page 196.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>zIIP Elig.</td>
<td>Step CPU Limit</td>
<td>zAAP CPU time</td>
<td>zIIP CPU time</td>
<td>Region Requested</td>
<td>Current use &lt;16M</td>
<td>Max used &lt;16M</td>
<td>Common Storage</td>
<td>Below</td>
<td>Above</td>
<td>MEMLIMIT</td>
<td>Current</td>
<td>High-Water Mark</td>
</tr>
<tr>
<td>0.00%</td>
<td>1,440.00 Min</td>
<td>0.00 Sec</td>
<td>0.00 Sec</td>
<td>7,144K</td>
<td>584K</td>
<td>988K</td>
<td>---CSA---</td>
<td>23,648</td>
<td>3,788K</td>
<td>NOLIMIT</td>
<td>0M</td>
<td>OM</td>
</tr>
<tr>
<td>1.20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,312</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. job ID of the specified address space
2. address space instance (asi)
3. indicates whether the address space is reusable
   - In the future, additional text might appear on this line for exceptional conditions related to the address space.
4. job message class and allocation and termination message levels
5. name of the program that was given control when the current step began execution, and the programmer name (if any) from the job statement
6. job start date in Julian and Gregorian formats
7. time of day that the job was initiated, and the initiation job class
8. time of day that the current step began execution, step name, current step number, and total number of steps in the job
9. amount of CPU time, in seconds, used by the step in TCB mode, and the amount of CPU time, in seconds, used by the step in SRB mode
10. total amount of CPU time, in seconds, used by the current step, and the percentage of the step time that was executed on a zAAP or zIIP processor, if one was available
11. amount of CPU time, in seconds, allotted to the current step, and the percentage of allocated CPU time that has been used
12. total amount of zAAP or zIIP processor time used by the step
   - (This figure is included in the total step time.)
13 amount of virtual storage requested by the current step
K indicates 1024-byte increments. You might also see M, which indicates 1,048,576-byte increments (Megabytes).

14 amount of virtual storage below the 16-MB line currently in use, and the amount above the 16 MB line currently in use

15 highest amount of virtual storage below the 16 MB line used by the current step, and the highest amount above the 16 MB line used by the current step

16 three messages that form a matrix displaying the following information:

<table>
<thead>
<tr>
<th>Common storage</th>
<th>CSA</th>
<th>SQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below</td>
<td>amount of below-the-line storage in the Common Service Area (CSA) that is currently owned by this address space</td>
<td>amount of below-the-line storage in the System Queue Area (SQA) that is currently owned by this address space</td>
</tr>
<tr>
<td>Above</td>
<td>amount of above-the-line storage in the Common Service Area (CSA) that is currently owned by this address space</td>
<td>amount of above-the-line storage in the System Queue Area (SQA) that is currently owned by this address space</td>
</tr>
</tbody>
</table>

The next three messages display the following information for memory objects. These messages are omitted if the operating system does not support memory objects.

17 MEMLIMIT value for this address space--the amount of private above-the-bar storage that the address space can allocate (G indicates Gigabytes, and T indicates Terabytes); LVMO is an acronym for Large Virtual Memory Object
Also, the entity that set the limit value; several possible self-describing values can appear in this position.

18 current amount of private above-the-bar storage allocated by the address, and the total amount of above-the-bar storage that is currently hidden (which is in addition to the amount allocated)

19 High-Water Mark: the maximum amount of private above-the-bar storage allocated by this address space; # Objects: the current number of memory objects allocated by this address space

**Note**
The OLIST service displays information for individual memory objects.

20 LNKLST Set Name currently in use for the specified address space
The PRSM (PRS) service displays CPU busy percentage and processor usage by partition.

When invoked in a PR/SM environment, the PRSM service monitors and reports logical partition activity. The average logical and physical processor utilization is shown for each partition. CPU busy percentage is computed for the current partition.

**Note**

For compatibility with previous releases of MainView for z/OS, PCPU is retained as a valid alias for the PRSM service.

### Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>length of the sample period in seconds; the default sample period is 10 seconds, and the maximum period is 300 seconds</td>
</tr>
<tr>
<td>ALL</td>
<td>includes inactive as well as active partitions in the display</td>
</tr>
</tbody>
</table>

### Example

To display CPU busy percentage and processor usage when running under the IBM PR/SM complex, type

```
PRSM,10
```

```
AMTL01I Collecting statistics for 10 Seconds
AMTLP1I Partitions: 24; CPUS: CP 18, ICF 4

<table>
<thead>
<tr>
<th>AMTLP2I</th>
<th>AMTLP3I</th>
<th>AMTLP4I</th>
<th>AMTLP5I</th>
<th>AMTLP6I</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJSC</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
<td>CP</td>
</tr>
<tr>
<td>CF01</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
</tr>
<tr>
<td>CF02</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
</tr>
<tr>
<td>CF03</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
<td>ICF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU</th>
<th>Log.</th>
<th>W</th>
<th>A</th>
<th>Relative</th>
<th>% Log.</th>
<th>% Phy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICF</td>
<td>1</td>
<td></td>
<td></td>
<td>-DED-</td>
<td>100.0</td>
<td>----</td>
</tr>
<tr>
<td>ICF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICF</td>
<td>1</td>
<td>100</td>
<td></td>
<td>12.5</td>
<td>90.8</td>
<td>22.7</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>AMTLP6I</th>
<th>Processor Type: CP</th>
<th>[14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLP6I</td>
<td>Processor Type: ICF</td>
<td>[14]</td>
</tr>
<tr>
<td>AMTLP7I</td>
<td>Current CPU</td>
<td>[19]</td>
</tr>
<tr>
<td>AMTLP8I</td>
<td>Partition ID</td>
<td>[19]</td>
</tr>
</tbody>
</table>

**Legend:**

1. number of seconds for the sample
2. number of partitions that have been defined on this PR/SM complex
3. number of physical CPUs in the PR/SM complex, broken down by type
4. number of processors of type CP and ICF
5. name assigned to the logical partition
6. CPU type
7. number of logical CPUs that will run in this logical partition
8. wait assist status
   - Y indicates that wait assist is available.
   - A blank indicates that wait assist is *not* available.
partition capped or not capped

A Y indicates that the partition is not allowed to exceed its relative share of CPU time, even if CPU resources are available.

weight that the dispatcher for PR/SM will give to this logical partition

divided by the sum of the weights for all partitions in the plex

average logical CPU use that this logical partition is making of the processor complex

average physical CPU use that this logical partition is making of the processor complex

physical processor utilization across the plex, grouped by type of processor

percentage of time that this type of processor was gainfully employed

percentage of this type of processor’s time that was used to determine how the processors should be dispatched to the various LPARs

percentage of time that this type of processor was not used

partition within which MainView SYSPROG Services is executing; also known as the host partition or host LPAR

More information is available for the host LPAR than for the other LPARs.

processor ID; one line for each physical processor that is assigned to the LPAR

percentage of time that the processor was busy while dispatching to the LPAR during the sample period; one line for each physical processor assigned to the LPAR

PSTORAGE

The PSTORAGE (PST) service provides the ability to determine the amount of private storage allocated by the various tasks within an address space.

It also provides the ability to determine

- the amount of private storage allocated to each subpool associated with a task
- the amount of private storage allocated to each key within a subpool associated with a task
Through the use of symbol names, PSTORAGE also provides the address and length of each allocated block. You can then use the DUMP service to display the contents of allocated storage. This procedure can be useful when you want to isolate the reason why private storage is not being freed properly and, therefore, is accumulating over time.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>
| symbol    | symbolic name in the form @TCB nnn  
The PSTORAGE service creates a symbolic definition of each task and displays the symbols along with other information when you enter an address space instance as the first parameter. Therefore, after you have displayed the tasks within an address space, you can reference each task symbolically.  
**Note:** The TCB service also creates symbolic names of this form. |
| spn       | subpool number (0-255)  
If you have previously referenced a task by its symbolic name, you can display subpool information for that task by specifying just the subpool number. If you want to display the subpool information that is associated with a different task, you can enter the symbolic name for the task, followed by the desired subpool number. |

**Note**

You can use PSTORAGE without parameters, but only after using PSTORAGE with an operand so that the @PVTxxx type symbols are available.

**Definitions**

**Allocated**

Storage is allocated to a specific subpool and key in page increments. Some of it might be free (available); the balance has been obtained.
Fragmented Free

Storage that has been *allocated* but not *obtained* is called *fragmented free storage*. This storage can be used only to satisfy a future *obtain* request for the same task, subpool, and key.

Obtained

Storage is *obtained* (GETMAINed) from storage that is *allocated* to the subpool and key.

Owned

Private storage is *owned* by a specific task. All private storage still owned by a task is automatically freed by the Recovery/Termination Manager when the task terminates.

Shared

Storage *owned* by one task can be shared with its subtasks. That is, if a subtask allocates storage in a shared subpool that it does not own and does not free the storage prior to termination, the storage remains *allocated* (and available for use) until the task that owns the subpool terminates.

*Note*

To avoid double-counting shared storage, the address space display excludes unowned storage from the total storage for a task. Some tasks might have zero storage because all of the storage they have allocated is within a subpool that is owned by a higher-level task. The amount of storage and its ownership status is displayed for each subpool in the task display and the subpool display.

### Example 1

To discover how much private storage is allocated by the GRS address space, type `PST,GRS`

<table>
<thead>
<tr>
<th>PSTORAG</th>
<th>AMTPQRI</th>
<th>Symbol</th>
<th>Adrs</th>
<th>Mod Name</th>
<th>Allocated</th>
<th>Obtained</th>
<th>Frag. Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTORAG</td>
<td>AMTPQ1I</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ2I</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB001</td>
<td>7FEOA8</td>
<td>IEAVAR00</td>
<td>126,976</td>
<td>47,736</td>
<td>79,240</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB002</td>
<td>7FFBF8</td>
<td>IEERW12</td>
<td>237,387,776</td>
<td>237,296,632</td>
<td>91,144</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB003</td>
<td>7CEEB8</td>
<td>ISGDRVR</td>
<td>65,536</td>
<td>65,536</td>
<td>0</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB004</td>
<td>7FF6F8</td>
<td>ISGDRVR</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB005</td>
<td>7FF918</td>
<td>ISGCMDRT</td>
<td>8,192</td>
<td>8,192</td>
<td>0</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB006</td>
<td>7FF6E8</td>
<td>ISGCMDRT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB007</td>
<td>7F54B0</td>
<td>ISGTRSMT</td>
<td>135,168</td>
<td>131,184</td>
<td>3,984</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ3I</td>
<td>@TCB008</td>
<td>7FEE8B</td>
<td>IEAVTSMD</td>
<td>69,632</td>
<td>59,272</td>
<td>10,360</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ2I</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>AMTPQ4I</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>PSTORAG</td>
<td>Total</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>237,793,280</td>
<td>237,608,552</td>
<td>184,728</td>
</tr>
</tbody>
</table>
**Note**
Tasks @TCB004 and @TCB006 have zero-allocated storage because this example shows only storage owned by the task. These tasks are using storage in a subpool that is owned by another task.

### Example 2

After displaying the amount of private storage owned by each task in the GRS address space, you want to know more about the storage that is owned by the task with the symbolic name @TCB002, so you type

```
PST, @TCB002
```

<table>
<thead>
<tr>
<th>AMTP05I</th>
<th>GRS</th>
<th>IEPRW12</th>
<th>@TCB002 at 7FFBF8</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTP06I</td>
<td>SPN</td>
<td>Allocated</td>
<td>Obtained</td>
</tr>
<tr>
<td>AMTP07I</td>
<td>---</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>AMTP08I</td>
<td>0</td>
<td>4,096</td>
<td>472</td>
</tr>
<tr>
<td>AMTP08I Own</td>
<td>229</td>
<td>724,992</td>
<td>701,520</td>
</tr>
<tr>
<td>AMTP08I Own</td>
<td>230</td>
<td>90,112</td>
<td>48,672</td>
</tr>
<tr>
<td>AMTP08I Own</td>
<td>252</td>
<td>397,312</td>
<td>387,632</td>
</tr>
<tr>
<td>AMTP09I</td>
<td>---</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>AMTP09I Total</td>
<td>237,387,776</td>
<td>237,296,632</td>
<td>91,144</td>
</tr>
</tbody>
</table>

**Note**
The O/S after the message number on the first line that contains message number AMTP08I means owned and shared. This task owns subpool zero and is sharing it with its subtasks.

### Example 3

After displaying the subpool information for task @TCB002 you want to know more about the storage that is allocated to subpool 127 for this task. Because you just displayed information for task @TCB002 and need to enter only the subpool number, type

```
PST, 127
```

<table>
<thead>
<tr>
<th>AMTP05I</th>
<th>GRS</th>
<th>IEPRW12</th>
<th>@TCB002 at 7FFBF8, SPN=127 Owned</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTP06I</td>
<td>Key</td>
<td>Allocated</td>
<td>Obtained</td>
</tr>
<tr>
<td>AMTP07I</td>
<td>---</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>AMTP08I Key</td>
<td>0</td>
<td>236,171,264</td>
<td>236,158,336</td>
</tr>
<tr>
<td>AMTP08I Key</td>
<td>Below the Line Storage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTP07I</td>
<td>---</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>AMTP08I Total</td>
<td>236,171,264</td>
<td>236,158,336</td>
<td>12,928</td>
</tr>
</tbody>
</table>

**Note**
All of the subpool 127 storage is in key zero and is allocated above the 24-bit address line.

### Example 4

You next decide to look at the storage that is allocated to task @TCB004, so you type
PST,@TCB004

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>GRS</th>
<th>ISGQDRVR</th>
<th>@TCB004 at 7FF6F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>SPN</td>
<td>Allocated</td>
<td>Obtained</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>Shr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTPQ8I</td>
<td>Total</td>
<td>4,096</td>
<td>472</td>
</tr>
</tbody>
</table>

**Note**

This storage is shared with, but not owned by, this task. Also, the storage is below the line.

**Example 5**

You now decide to find out more about the subpool 229 storage that is associated with task @TCB002. The last task that you referenced was @TCB004, and you must enter the symbolic task name in addition to the subpool number, so you type PST,@TCB002,229

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>IEPRW12</th>
<th>@TCB002 at 7FFBF8, SPN=229 Owned</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>Key</td>
<td>Allocated</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>0</td>
<td>552,960</td>
</tr>
<tr>
<td>AMTPQ8I</td>
<td>Below the Line Storage</td>
<td>---</td>
</tr>
<tr>
<td>AMTPQ9I</td>
<td>0</td>
<td>172,032</td>
</tr>
<tr>
<td>AMTPQ9I</td>
<td>Total</td>
<td>724,992</td>
</tr>
</tbody>
</table>

**Example 6**

You are now interested in knowing the size of each allocated block of subpool 229 storage that is associated with this task. You need to enter the PSTORAGE command without any operands, so you type PST

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>Label</th>
<th>Start</th>
<th>End</th>
<th>Pages</th>
<th>SPN</th>
<th>Ky</th>
<th>TCB#</th>
<th>Owner</th>
<th>Jobname</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>------</td>
<td>-----------</td>
<td>---------</td>
<td>------</td>
<td>-----</td>
<td>----</td>
<td>-------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT001</td>
<td>005CF000</td>
<td>005DBFFF</td>
<td>13</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT002</td>
<td>005DC000</td>
<td>005DFFFF</td>
<td>4</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT003</td>
<td>005E0000</td>
<td>005F4FFF</td>
<td>21</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT004</td>
<td>005F8000</td>
<td>005FBFFF</td>
<td>4</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT005</td>
<td>7F838000</td>
<td>7F84CFFF</td>
<td>21</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT006</td>
<td>7F852000</td>
<td>7F8A2FFF</td>
<td>81</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT007</td>
<td>7F8A3000</td>
<td>7F8B7FFF</td>
<td>21</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT008</td>
<td>7FFD0000</td>
<td>7FFD0FFF</td>
<td>4</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>@PVT009</td>
<td>7FFD2000</td>
<td>7FFD9FFF</td>
<td>8</td>
<td>229</td>
<td>0</td>
<td>2</td>
<td>5FDE28</td>
<td>GRS</td>
</tr>
</tbody>
</table>

**Example 7**

You notice that the third allocation block is 21 pages and are interested in its contents. The PSTORAGE service assigned label @PVT003 to this block, so you use the DUMP service to display the beginning of this area and type DU,@PVT003

AMTC12I 005E0000 08 C7E5E3E7 00000000 00000000 00000000 *GVTX............*
AMTC12I 005E0010 00000000 8002C248 80029B88 00000000 *......B.........*
AMTC12I 005E0020 8002B1A8 8002A92E 00000000 8002A934 *...y....z....z.*
AMTC12I 005E0030 8002A93A 8002A940 00005A18 8001C508 *...z....z....EQ*
AMTC12I 005E0040 00000000 00000000 00000000 9AF02720 *................O.*
AMTC12I 005E0050 00000000 00000000 00000000 00000000 *........................*
Example 8

You would like to see more data, so you first enter "DUMP +80,128" to display the next 128 bytes, then repeat that command by entering only an asterisk (*) to step through the storage:

```
DU,*,+80,128
```

The QUEUE (QUE) service displays current enqueue conflicts or requested system enqueue information.

Several filter options are provided, including the ability to filter by using a job name.

**Note**

The QUEUE service functionally replaces the ENQUEUE service, while providing new features. The ENQUEUE service is still available, but BMC recommends the use of the QUEUE service over the ENQUEUE service.

A conflict exists when one or more requesters are waiting for a resource. If this occurs frequently for a given resource, you can cause that resource to be omitted from the display by adding it to BBPARM member $$XENQ. See "$XENQ" on page 32.

The QUEUE service permits the use of several parameters, the defaults of which have been designed to simplify the use of the service. For example, you can display current enqueue conflicts by omitting all parameters, or display enqueue information for a specific data set name by specifying only the data set name. You can also display enqueue information for multiple data sets whose names match the pattern you have entered.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNAME=rname</td>
<td>RNAME for the search</td>
</tr>
<tr>
<td></td>
<td>Specify a 1- to 64-character literal value (see the explanation of literal value in the section “Definitions” on page 210). Optionally, the keyword RNAME can be omitted if you enter the RNAME value in the first parameter position. The literal value can contain up to 64 characters and can contain pattern characters. See the description of pattern characters in the section “Definitions” on page 210.</td>
</tr>
<tr>
<td>QNAME=qname</td>
<td>identifies the QNAME value for the search</td>
</tr>
<tr>
<td></td>
<td>Specify a one- to eight-character literal value (see the explanation of literal value in the section “Definitions” on page 210). Optionally, the keyword QNAME= can be omitted if you enter the QNAME value in the second parameter position. If omitted, QNAME='SYSDSN' is assumed unless JOBNAME= or TASK= is specified, in which case the default is QNAME='. The QNAME value can contain pattern characters. See the description of pattern characters in the section “Definitions” on page 210.</td>
</tr>
<tr>
<td>JOBNAME=jobname</td>
<td>(optional) when specified, limits the returned information to the specified address space</td>
</tr>
<tr>
<td></td>
<td>Optionally, the keyword JOBNAME= can be omitted if you enter the JOBNAME value in the third parameter position. See the explanation of asi in the section “Identifying address spaces” on page 47 for the various forms that can be used to identify the address space. The JOBNAME parameter is mutually exclusive with the TASK= parameter.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SCOPE=keyword</td>
<td>defines the SCOPE of the resources that are to be returned. The following</td>
</tr>
<tr>
<td></td>
<td>values are allowed: ANY, SYSTEM, SYSTEMS, and SYSPLEX. The default is ANY.</td>
</tr>
<tr>
<td>TASK=symbolName</td>
<td>(optional) specifies a symbol that identifies the desired task that was</td>
</tr>
<tr>
<td></td>
<td>created by the TCB or PSTORAGE service. The TASK parameter is mutually</td>
</tr>
<tr>
<td></td>
<td>exclusive with the JOB parameter.</td>
</tr>
<tr>
<td>RTOKEN</td>
<td>(optional) indicates that the 32-byte resource token is to be displayed for</td>
</tr>
<tr>
<td></td>
<td>each listed resource. You can specify NO or YES. The default is NO.</td>
</tr>
<tr>
<td></td>
<td>RTOKEN=YES is supported only on IBM z/OS 1.9 and later.</td>
</tr>
</tbody>
</table>

**Note**

- All keyword parameters can be abbreviated to their first two letters. For example, you can specify QNAME= or QN=.
- RNAME and QNAME values are displayed in hexadecimal if they contain nonprintable characters.
- RTOKEN values are displayed in hexadecimal.

**Definitions**

**Literal values**

Within the context of the QNAME and RNAME parameters of the QUEUE service, you can enter a literal value in the following ways:

<table>
<thead>
<tr>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>abcd</td>
<td>The value is not enclosed within quotation marks so it is converted to</td>
</tr>
<tr>
<td></td>
<td>uppercase. The result is 'ABC'.</td>
</tr>
<tr>
<td>‘abcd’</td>
<td>The value is enclosed in quotation marks so it is assumed to be a character</td>
</tr>
<tr>
<td></td>
<td>value and is not converted to uppercase. Therefore, the result is 'abcd'.</td>
</tr>
<tr>
<td>C‘abcd’</td>
<td>The value is enclosed in quotation marks so it is assumed to be a character</td>
</tr>
<tr>
<td></td>
<td>value and is not converted to uppercase. Therefore, the result is 'abcd'.</td>
</tr>
<tr>
<td>X‘abcd’</td>
<td>The value begins with an X followed by a quotation mark and ends with a</td>
</tr>
<tr>
<td></td>
<td>quotation mark; therefore, it is considered to be a two-byte hexadecimal</td>
</tr>
<tr>
<td></td>
<td>value. The case is ignored.</td>
</tr>
</tbody>
</table>
Example Explanation

abcd(64) The suffix (64) indicates that the preceding abcd is an expression that, when evaluated, is the address of the value in storage. The decimal number enclosed within parentheses is the length of the value.

Therefore, if symbol ABCD has been previously defined, then the value of that symbol (and its address space attribute), when evaluated, determines the start of the value within storage. In this example, the expression ABCD could also be the hexadecimal address X’ABCD’. If the symbol ABCD does not exist, it is assumed to be an address. If the first term in the expression is not a defined symbol and contains nonhexadecimal characters, it is considered invalid.

The length of this value can be up to 255 bytes. See Expressions on page 48 for additional information.

Pattern characters

You can use the ? and * pattern characters within the QNAME and RNAME parameters.

- ? indicates that any character can appear in that position.
- * indicates that any number of characters (including zero) can appear in that position.

All other characters must match for the enqueue to be selected for display. For example, a pattern of ABC*DEF would match on any of the following values:

- ABCDEF
- ABC123DEF
- ABC.DEF

Example 1

To list all enqueue information for data sets beginning with SYS2.BB, type

**QUE,SYS2.BB**

<table>
<thead>
<tr>
<th>AMTQU1I</th>
<th>Your search located 21 resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQU2I</td>
<td>Scope=SYSTEM  Owners= 3  No Waiters</td>
</tr>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=SYS2.BBX.BBLINK</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID  Jobname  ASID  TCB  Stat  Typ  Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  XCFAS  0006  6FF890  Owns  Shr  22:16:41 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  LLA  001B  6FF890  Owns  Shr  22:16:42 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$DSTR  007F  6FF890  Owns  Shr  22:20:01 5/22/2012</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTQU2I</th>
<th>Scope=SYSTEM  Owners= 8  No Waiters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=SYS2.BBPROD.BBPARM</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID  Jobname  ASID  TCB  Stat  Typ  Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$BBID  0021  6FF890  Owns  Shr  22:19:04 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$CAS  002C  6FF890  Owns  Shr  22:19:07 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$PAS  002D  6FF890  Owns  Shr  22:19:07 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$TOM  004E  6FF890  Owns  Shr  22:19:42 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$RES  007B  6FF890  Owns  Shr  22:20:02 5/22/2012</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD  DC$DSTR  007F  6FF890  Owns  Shr  22:20:01 5/22/2012</td>
</tr>
</tbody>
</table>
Example 2

To display enqueue information for a specific data set, including the resource token, type

QUE,VAM3.GEN.LOAD

Your search located 1 resource

Note

In this example, R= was omitted because it is optional when specified in the first parameter position.

Example 3

The TCB service assigns a symbolic name to each task which may be used to identify a task to the QUEUE service. Also, the QUEUE service uses these symbolic names in preference to the actual TCB address when they are present.

To display all enqueues held by a specified task, type

QUE,TASK=@TCB2

Your search located 152 resources
**Note**
The TCB service was used to assign symbolic names to the address space's tasks.

**Example 4**

To display ENQ conflicts, type

```que
QUEUE
```

```amtq
AMTQU1I Your search located 2 resources
AMTQU2I Scope=SYSTEM Owners= 1 Waiting for: Excl= 1
AMTQU3I Q=SYSZJES2
AMTQU4I R=SJB.1A245728
AMTQU5I SYSID  Jobname ASID TCB Stat Typ
AMTQU6I SJSD  VAM3ENQ  62 @TCB003 Owns Exc 15:02:56 4/22/YYYY

AMTQU2I Scope=SYSTEM Owners= 1 Waiting for: Excl= 1
AMTQU3I Q=SYSZJES2
AMTQU4I R=SJB.1A245728
AMTQU5I SYSID  Jobname ASID TCB Stat Typ
AMTQU6I SJSD  VAM3ENQ  62 @TCB003 Owns Exc 15:02:56 4/22/YYYY
```

Chapter 4 Services 213
The REPLIES (REP) service displays outstanding Write To Operator with Reply (WTOR) messages.

Use the REPLIES service to ensure that these messages are not overlooked.

**Syntax**

```
REPLIES systemName
REPLIES ALL
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemName</td>
<td>one- to eight-character system name</td>
</tr>
<tr>
<td>ALL</td>
<td>displays replies from all systems</td>
</tr>
</tbody>
</table>

If you type `REPLIES` without parameters, only messages from the current system are displayed.

**Note**

You can exclude messages issued by specific address spaces (see “$$RPJOBS” on page 29) or containing specific text (see “$$RPTEXT” on page 31).
Example 1

To display outstanding replies from the current system, type

REP

<table>
<thead>
<tr>
<th>Name or ID of the system where the replies were issued</th>
<th>JES job ID, including the job number and type (JOB for batch job, STC for Started Task, and TSU for time-sharing user)</th>
<th>Reply number</th>
<th>Reply text</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJSC</td>
<td>@1261 DBC640Q (BMCETHS3) XDC S2.0 DEBUGGER</td>
<td>1</td>
<td>C TO CANCEL WAIT</td>
</tr>
<tr>
<td>SJSC</td>
<td>I7ATPNSC @1260 ITP001E 10.49.47.47 TPNS I7ATPNSC</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SJSC</td>
<td>IMS61X *1233 DFS996I <em>IMS READY</em> X19H</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SJSC</td>
<td>BOLLXS2 @1229 BBWIA002A MvExplorer Host Server is ready</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SJSC</td>
<td>I7AM38CT *1146 DFS996I <em>IMS READY</em> X17A</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SJSC</td>
<td>DC$ADMC @1142 ADM2000 I ADMOPUT(DC$ADMC). TO TERMINATE, REPLY 'STOP', 'STOPQ', OR 'STOPS'</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
</tbody>
</table>

Example 2 – MainView AutoOPERATOR output

To display outstanding replies from the current system, type

REPLIES

<table>
<thead>
<tr>
<th>Name or ID of the system where the replies were issued</th>
<th>JES job ID, including the job number and type (JOB for batch job, STC for Started Task, and TSU for time-sharing user)</th>
<th>Reply number</th>
<th>Reply text</th>
</tr>
</thead>
<tbody>
<tr>
<td>STC03223</td>
<td>*94 DSI802A CNM01 REPLY WITH VALID NCCF</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SYSB</td>
<td>SYSTEM OPERATOR COMMAND</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>STC01830</td>
<td>*60 DFS996I <em>IMS READY</em> X15H</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>SYSB</td>
<td>ADMOPUT(ADMPRINT). TO TERMINATE, REPLY 'STOP', 'STOPQ', OR 'STOPS'</td>
<td>1</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
</tbody>
</table>

Legend:

1 name or ID of the system where the replies were issued
2 JES job ID, including the job number and type (JOB for batch job, STC for Started Task, and TSU for time-sharing user)
3 reply number
4 reply text

RESERVES

The RESERVES (RES) service helps determine the cause of poor DASD performance or system lockouts.

It also

- displays reserve activity for DASD devices
- displays system contention caused by long-term reserves

A reserve issued by one processor for a single resource prevents all other processors from accessing the entire volume.

**Note**

To prevent reserve information for specific volumes from displaying, add the VOLSER to BBPARM member $$XRES. See “$$XRES” on page 33.

**Syntax**

```
RESERVES
```

**Example**

To display reserve activity on DASD devices, type

```
RESERVES
```

<table>
<thead>
<tr>
<th>AMTQ2PI</th>
<th>SYSTEM</th>
<th>(LOCAL)</th>
<th>Q=PINION</th>
<th>R=BAB301</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQ2QI</td>
<td>SYSID</td>
<td>JOBNAME</td>
<td>ASID</td>
<td>STAT TYP TIME 15:01:14</td>
</tr>
<tr>
<td>AMTQ2RI</td>
<td>SJSD</td>
<td>VAMSRES</td>
<td>(0057) OWNS EXC RES=001 877B BAB301</td>
<td></td>
</tr>
<tr>
<td>AMTQ1LI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. scope of enqueue; possible scopes are as follows:
   - SYSTEMS
   - SYSTEM
   - STEP

2. whether the resource is global or local

3. major name (Q=)

4. minor name (R=)

5. system ID for the system executing the task that is holding or waiting for the resource

6. job name, TSO user ID, or Started-Task ID for the address space containing the task that is holding or waiting for the resource

The job name is not available in some situations.
7 ID for the address space holding or waiting for the resource

8 current status (OWN$ or WAIT), indicating that the task holds (OWN$) the resource or is waiting (WAIT) for the resource

9 type of enqueue: EXC for exclusive, SHR for shared

10 reserve information

   - RES indicates that a reserve is associated with the enqueue. The reserve count is also displayed unless it is zero, in which case the equal sign is also omitted.

   - CVT indicates that reserve request has been converted to global enqueue. The remaining four fields may not appear.

11 if present, indicates that the reserve is pending, which means that the task is waiting to reserve the devices
   Generally, the device is reserved by another system.

12 device address in hexadecimal format

---

**RSM**

The RSM service provides information about the allocation of real storage within the system.

This service also determines the number of pageable and fixed real-storage frames allocated to system common storage areas (LPA, CSA), the system queue area (SQA), and private storage. You can also display the number of frames allocated to each address space.

You can use this information to observe the

- effects of real-storage page fixing
- usage of real-storage frames residing above the 16 MB line
- impact of specific workloads on the real-storage resource
## Syntax

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>provides a tabular display of the amount of real storage allocated to each address space</td>
</tr>
<tr>
<td>ASID</td>
<td>sorts the output in ascending order by address space identifier</td>
</tr>
<tr>
<td>NAME</td>
<td>sorts the output in alphabetical order by address space name or system area</td>
</tr>
<tr>
<td>FRAMES</td>
<td>sorts the output in descending order by the total number of frames allocated to the address space or system area; only the address spaces with the 10 highest numbers of allocated frames are displayed</td>
</tr>
<tr>
<td>FIXED</td>
<td>sorts the output in descending order by the total number of frames allocated to the address space or system area that are marked as fixed; only the address spaces with the 10 highest numbers of allocated fixed frames are displayed</td>
</tr>
<tr>
<td>&lt;16MB</td>
<td>sorts the output in descending order by the number of frames allocated to the address space or system area that are currently marked as fixed below the 16-MB line; only the address spaces with the 10 highest number of fixed frames below the 16-MB line are displayed</td>
</tr>
<tr>
<td>LSQA</td>
<td>sorts the output in descending order by the total number of frames allocated to the local system queue area for the listed address space; only the address spaces with the 10 highest number of frames allocated to the local system queue area are displayed</td>
</tr>
<tr>
<td>WSS</td>
<td>sorts the output in descending order by the working set size of the address space at the time of the last swap-out; only the address spaces with the 10 largest working set sizes are displayed</td>
</tr>
<tr>
<td>PERCENT</td>
<td>sorts the output in descending order by the percentage of online real-storage frames allocated to the address space; only the address spaces with the 10 highest percentages of online real-storage frame allocations are displayed</td>
</tr>
</tbody>
</table>
Example 1

To display real-storage allocation, type

RSM

<table>
<thead>
<tr>
<th>AMTRSI</th>
<th>Total</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTRSI1 Private frames:</td>
<td>222157</td>
<td>27295</td>
</tr>
<tr>
<td>AMTRSI2 Common frames:</td>
<td>18088</td>
<td>568</td>
</tr>
<tr>
<td>AMTRSI3 SQA frames:</td>
<td>10748</td>
<td>10748</td>
</tr>
<tr>
<td>AMTRSI4 Free frames:</td>
<td>4713</td>
<td></td>
</tr>
<tr>
<td>AMTRSI5 Total online:</td>
<td>255706</td>
<td>38611 (16MB 693)</td>
</tr>
<tr>
<td>AMTRSI6 Nucleus:</td>
<td>2323</td>
<td>2323</td>
</tr>
<tr>
<td>AMTRSI7 PFT frames:</td>
<td>4109</td>
<td></td>
</tr>
<tr>
<td>AMTRSI8 <em>Unknown</em>:</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>AMTRSI9 Total:</td>
<td>262144 (1024M)</td>
<td>40934</td>
</tr>
</tbody>
</table>

Legend:

1 total frames allocated to private areas

2 fixed frames allocated to private areas

3 total RCECOMAL-SQA frames allocated to common areas (LPA+CSA)

4 fixed frames allocated to common areas (CSA+LPA)

5 frames allocated to SQA (source: z/OS RSM)

6 always same as SQA total

7 total free frames (calculation: RCEAFC)

8 always blank

9 number of frames currently online (calculation: RCEPOOL is equal to the total of the fields in the column)

10 total fixed frames allocated in system (calculation: RCETOTFX is equal to the total of the fields in the column)

11 frames allocated to system nucleus

12 same as nucleus total

13 number of frames used by the Page Frame Table

14 always blank

15 frames unaccounted for (bad, offline)
16 total storage on the machine

17 total number of fixed frames

**Example 2**

To display the ten address spaces that are using the most real storage in descending sequence, as well as the use of real storage by the common areas, type

```
RSM,PERCENT
```

<table>
<thead>
<tr>
<th>AMTR501</th>
<th>Total</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTR511 Private Frames:</td>
<td>218716</td>
<td>29085</td>
</tr>
<tr>
<td>AMTR521 Common Frames:</td>
<td>14476</td>
<td>630</td>
</tr>
<tr>
<td>AMTR531 SQA Frames:</td>
<td>16243</td>
<td>16243</td>
</tr>
<tr>
<td>AMTR541 Free Frames:</td>
<td>6262</td>
<td></td>
</tr>
<tr>
<td>AMTR551 Total Online:</td>
<td>255697</td>
<td></td>
</tr>
</tbody>
</table>

| AMTR561 Nucleus: | 2332 | 2332 |
| AMTR571 PFT Frames: | 4109 | |
| AMTR581 *Unknown*: | 6 | |
| AMTR591 Total: | 262144 (1024M) | 48290 |

```
AMTR25I NAME ASID FRAMES FIXED < 16MB LSQA WSS PERCENT
AMTR26I * FREE * ***** 6262 0 0 0 2
AMTR26I SQA ***** 16243 16243 0 0 0 6
AMTR26I LPA ***** 3840 125 0 0 0 1
AMTR26I CSA ***** 10636 505 32 0 0 4
AMTR26I NUCLEUS ***** 2332 2332 0 0 0 0
```

**Note**

The AMTR2AI line is not displayed when you are operating under MainView AutoOPERATOR.

**Legend:**

1 name of the address space or system area whose frame counts are being provided

The first entry (flagged as *FREE*) contains a count of the number of unallocated frames in the system.

The system queue area (SQA), link-pack area (LPA), and common storage area (CSA) entries follow the FREE entry. The system area entries are followed by the entries for the system address spaces (*MASTER*, PCAUTH, and so on) that are established during system initialization.

Entries for each additional address space in the system are then listed (in ASID order).
2 hexadecimal address space identifier
   This field contains asterisks (****) for the free space and system entries.

3 total number of frames allocated to the address space or system area

4 total number of frames allocated to the address space or system area that are
   currently marked as fixed
   This value is a subset of the total frame count provided in the FRAMES field.

5 total number of frames allocated to the address space or system area that are
   currently marked as fixed below the 16-MB line
   This value is a subset of the total fixed-frame count provided in the FIXED field.

6 total number of frames allocated to the local system queue area for the listed
   address space
   This value is a subset of the total fixed-frame count provided in the FIXED field.

7 working set size of this address space at the time of the last swap-out
   This field is valid only for swappable address spaces. The nonsystem address
   spaces, which start out as swappable but later change status to nonswappable,
   continue to display what appears to be a fixed working set size. In this case, the
   count actually shows the working set size at the time the address space became
   nonswappable.

8 percentage of online real-storage frames allocated to this address space
   This value is calculated as follows:
   \[
   \frac{(Total\ frames\ (pageable\ and\ fixed)\ allocated\ to\ the\ address\ space\ or\ system\ area\ /\ RCEPOOL) \times 100\%}{\}
   \]
   If the result of this calculation yields a result less than 1%, the field is replaced by
   dashes (−).

SCT

The Set Counter (SCT) service provides the ability to enable/disable and activate/
deactivate one or more sampler sets.

Syntax

```
SCT
```

```
setname, action
```

```
setname, action
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| setname   | is one of the following values:  
  - PROBLEM-STATE  
  - BASIC  
  - EXTENDED  
  - COPROCESSOR  
  - CRYPTO-ACTIVITY |
| action    | is one of the following values:  
  - ENABLE  
  - DISABLE  
  - ACTIVATE  
  - DEACTIVATE |

One or more pairs of parameters can be specified. setname and action keywords can be truncated to the first three characters.

ACTIVATE implies ENABLE, and DISABLE implies DEACTIVATE. Therefore, customers are not required to ENABLE prior to activating a counter set.

**SDUMP**

The SDUMP (SDU) service takes an SVC dump of one or more address spaces.

The SDUMP service works in conjunction with the SYSDUMP service (which lists the SVC dump data sets). See “SYSDUMP” on page 251 for more information.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>
| title     | is the title of the dump  
This is an optional positional parameter. If this value is omitted, the SDUMP service generates a title containing relevant information, including the ASID of each dumped address space. 
BMC recommends that you not specify a value for this parameter. |
<p>| ALLNUC    | includes the DAT-ON and DAT-OFF nuclei in the display; these areas are not dumped by default |
| ALLPSA    | includes all of the prefixed storage areas (PSAs) in the system; these areas are not dumped by default |
| NUC       | includes the nonpage-protected areas of the DAT-ON nucleus in the display; these areas are not dumped by default |
| NOCSA     | suppresses display of the CSA and ECSA subpools; these areas are dumped by default |
| NOGRSQ    | suppresses display of the global resource serialization control blocks; these control blocks are dumped by default |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOLPA</td>
<td>suppresses display of the active link-pack area modules and SVCs; these areas are dumped by default</td>
</tr>
<tr>
<td>NOLSQA</td>
<td>suppresses display of the LSQA and ELSQA subpools; these subpools are dumped by default</td>
</tr>
<tr>
<td>NOPSA</td>
<td>suppresses display of the PSA for one processor, which is either the processor at the time of the error or the processor at the time of the dump; the PSA is dumped by default</td>
</tr>
<tr>
<td>NORGN</td>
<td>suppresses display of the allocated pages in the address space’s private area; these pages are dumped by default</td>
</tr>
<tr>
<td>NOSQA</td>
<td>suppresses display of the SQA and ESQA subpools; these subpools are dumped by default</td>
</tr>
<tr>
<td>NOSUMDUMP</td>
<td>suppresses display of a summary dump; this dump is included by default</td>
</tr>
<tr>
<td>NOSWA</td>
<td>suppresses display of the scheduler work area subpools for the address space; these subpools are dumped by default</td>
</tr>
<tr>
<td>NOTRT</td>
<td>suppresses display of the system trace table, the GTF trace records, and master track data, if these traces are active; these areas are dumped by default</td>
</tr>
</tbody>
</table>

**Note**

These areas are defined by IBM. For more information, see the IBM publication, *MVS Programming: Authorized Assembler Services Reference, Volume 3*.

**Example**

To obtain an SVC dump of three address spaces, type

```
SDUMP (.PAS,.CAS,(1))
```

```
AMTDSD7I SVC dump in progress
AMTDSTI Token='BBPLEX01 SJSD     10/29/2013 17:50:00'

AMTDSD2I COMPLETE dump has been taken for:
AMTDSDJI ASID Fully Qualified Name
AMTDSDKI ---- -------------------
AMTDSDBI 002F DC$PAS.PAS
AMTDSDBI 002E DC$CAS.CAS
AMTDSDBI 0001 *MASTER*
AMTD001A SYSPROG
```

- The names of the address spaces are enclosed within parentheses, indicating that they are subparameters of the first parameter.
- The third address space is defined by its ASID, instead of its name, by enclosing the ASID within parentheses. The ASID is assumed to be a decimal value unless preceded by an X, which would indicate a value in hexadecimal (for example, X1).
- No dump options were specified (third and subsequent parameter positions); therefore, the default dump options are used.
The SETOP (SET) service allows you to set options specific to your session.

You can display the current option settings by typing **SETOP** without any operands or change an option by specifying it as a SETOP parameter.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGID</td>
<td>This parameter indicates that message numbers are to be displayed at the beginning of messages produced by MainView SYSPROG Services.</td>
</tr>
<tr>
<td>NOMSGID</td>
<td>This parameter indicates that message numbers should not be displayed on messages produced by MainView SYSPROG Services.</td>
</tr>
<tr>
<td>CAPSON</td>
<td>This parameter indicates that messages are to be displayed in uppercase.</td>
</tr>
<tr>
<td>CAPSOFF</td>
<td>This parameter indicates that messages are to be displayed in mixed case.</td>
</tr>
</tbody>
</table>
| DUMP      | This parameter indicates that an SVC dump should be taken if a service that you invoke terminates abnormally.  
The DUMP=NO | YES option in BBPARM member $$INSYS0 controls the default. NODUMP is the default unless DUMP=YES is specified in $$INSYS0. |
| NODUMP    | This parameter indicates that an SVC dump should *not* be taken if a service that you invoke terminates abnormally.  
The DUMP=NO | YES parameter in BBPARM member $$INSYS0 controls the default. NODUMP is the default unless DUMP=YES is specified in $$INSYS0. |
| ECHO      | This parameter indicates that the commands you enter should be recorded in the PAS job log. The ECHO=NO | YES parameter in BBPARM member $$INSYS0 controls the default. NOECHO is the default unless ECHO=YES is specified in $$INSYS0. This parameter is applicable only when you are access MainView SYSPROG Services using the MainView PAS. |
### Parameter

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>NOECHO</td>
<td>This parameter indicates that the commands you enter should <em>not</em> be recorded in the PAS job log. The ECHO=NO</td>
</tr>
<tr>
<td>RWAIT= $nn$</td>
<td>This parameter specifies the maximum amount of time, in seconds, that the COMMAND service will wait for a command response. $nn$ can be any decimal value from 0 to 15; the default is 3. The global parameter value can be set using the RWAIT parameter in PARMLIB member $$INSYS0. If the parameter is not specified, the default is three seconds.</td>
</tr>
<tr>
<td>OUTLIM= $n$</td>
<td>When MainView SYSPROG Services is operating under MainView, this parameter specifies the maximum number of output lines a service can produce prior to a prompt being issued asking if more output is desired. $n$ can be any value within the range 100 to 25000. The default is 1000.</td>
</tr>
<tr>
<td>NOROLL= $n$</td>
<td>This parameter provides the ability to change the value established by the AEWNOROLL parameter in BBPARM library member $$INSYS0. See the description of AEWNOROLL in Table 3 on page 25.</td>
</tr>
</tbody>
</table>

### Example

To display the current options, type

```
SET AMTS01I MSGID CAPSOFF DUMP NOECHO RWAIT=10
AMTS02I OUTLIM= 1,000
AMT001A SYSPROG
```

### SMFEXIT

The SMFEXIT (SMF) service loads, enables, disables, and deletes SMF exit routines and displays the status of SMF exits.

Use this service to add or delete SMF exit routines dynamically.

An *exit point* (also called simply an exit) calls the exit routine. The exit point is uniquely identified by a combination of the SMF exit name and the subsystem name. Multiple exit routines can be loaded and enabled for a given exit point.

An *exit routine* is a load module that is called by an exit point. The exit point can establish requirements for the exit routine. The exit can require that the exit routine be re-entrant or that it be in a certain addressing mode. All exit routines must be loaded from an APF-authorized data set. The same exit routine can be used with
multiple exit points if designed for this purpose. BMC Software recommends that user-written or third-party exit routines be given names that do not duplicate existing IBM exit names. For example, use XYZUJV instead of IEFUJV.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| STATUS    | displays the status of all or selected SMF exits  
All exits are displayed unless you specify parameters to limit the display. For example:  
- If you specify the exit name, all exit routines for the specified exit name are displayed.  
- If you specify a subsystem name, all exit routines for the subsystem are displayed.  
- If you specify both an exit name and a subsystem name, only the exit routines for that subsystem and exit are displayed.  
- If you specify a module name, all exits using that module name are displayed. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD</td>
<td>causes the load module (exit routine) to be associated with the specified SMF exit point; the exit point is identified by the SMF exit name and the subsystem name. The operating system does not actually load the module until you ENABLE the exit. However, validation is performed by the SMFEXIT LOAD operation. If successful, the exit routine you have loaded will be loaded under DISABLE status. To activate the exit, you must use the ENABLE function. You can associate multiple exit routines with the same exit as long as they have different load module names. To reload the exit routine (MODNAME) for the same exit, you must first DISABLE and then DELETE the exit routine. New versions of an exit routine should be uniquely named so that the prior version can be retained (in a DISABLE status). This way, it can be reactivated (ENABLE) if the new exit routine is in error. You must specify the exit name, subsystem name, and load module name on the LOAD request. The data set name is also required unless previously specified.</td>
</tr>
<tr>
<td>ENABLE</td>
<td>causes the specified exit routine (MODNAME) to be activated for the specified exit point. Initially, modules are loaded in DISABLE status and therefore cannot be called by the exit point until you change their status to ENABLE. The SMF exit name and the subsystem name are combined to form an exit identifier that uniquely identifies the exit point. One or more exit routines can be associated with (called by) a single exit point (or exit). The exit routine is identified by its load module name (MODNAME).</td>
</tr>
<tr>
<td>DISABLE</td>
<td>changes the status of the exit routine for the specified exit to DISABLE. The exit routine remains in storage and can be changed to an ENABLE status again unless the DELETE function is used to delete the exit routine. You must specify the exit name, subsystem name, and load module name to use the DISABLE command to disable the exit routine.</td>
</tr>
<tr>
<td>DELETE</td>
<td>causes the exit routine to be removed from storage; the exit routine must be disabled with the DISABLE command before it can be deleted. You must specify the exit name, subsystem name, and load module name to use the DISABLE command to disable the exit routine.</td>
</tr>
<tr>
<td>exitName</td>
<td>name of the SMF exit that is to be used with the LOAD, ENABLE, DISABLE, or DELETE commands or that is to be displayed by the SMFEXIT service. Valid SMF exit names are ACTRT, UJI, UJP, UJV, USI, USO, UTL, U29, U83, and U84. The SMF exit name and the subsystem name are combined to form an exit identifier that uniquely identifies the exit point. One or more exit routines can be associated with (called by) a single exit point (or exit). The exit routine is identified by its load module name (MODNAME).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| subsys    | name of the subsystem  
This name is the subsystem name to be used to identify the exit point. For STATUS requests, the subsystem name can be specified individually or in combination with other parameters to limit the display.  
Valid subsystem names are STC, SYS, TSO, JES2, and JES3.  
The SMF exit name and the subsystem name are combined to form an exit identifier that uniquely identifies the exit point. One or more exit routines can be associated with (called by) a single exit point (or exit). |
| modname   | load module name for the exit routine  
An exit routine can be associated with (loaded and enabled for) multiple exit points if desired. An exit point is uniquely identified by an exit identifier that is a combination of the SMF exit name and the subsystem name.  
In addition, multiple exit routines (each with a unique load module name) can be loaded and enabled for a given exit point. However, multiple versions of an exit routine with the same load module name are not supported.  
The default module name is the IBM exit routine name. However, BMC Software recommends that you use a name other than the IBM name when creating an exit routine. |
| dsname    | name of the data set containing the exit routine (MODNAME) that is to be loaded  
When specified, the data set name is retained for the life of the MainView for z/OS session and, therefore, need not be specified again on subsequent LOAD requests.  
The data set must be a partitioned data set. It must be cataloged and it must be APF authorized. If it is not APF authorized, the MainView for z/OS APF service can be used to authorize it. |

**Examples**

To display all SMFEXIT routines that are associated with the SMF UJI exit, type

```
SMF STATUS UJI
```

To display SMFEXIT routines that are associated with the SMF UJI exit for subsystem STC, type

```
SMF STATUS UJI STC
```

To load exit routine MYUJV from SYS1.LPALIB for the SMF UJV exit and subsystem SYS, and then load the same exit routine for subsystem STC, type

```
SMF LOAD,UJV,SYS,MYUJV,SYS1.LPALIB
SMF LOAD,UJV,STC,MYUJV
```

*Note*

The second command is identical to the first one except that it is for subsystem STC. The data set name is required only on the first load command.

To enable the exit routine that was previously loaded, type

```
SMF ENABLE,UJV,SYS,MYUJV
```
**Note**
Exit routines are disabled when loaded and must be enabled.

To disable the exit routine for the specified SMF exit and subsystem, type

```
SMF DISABLE,UJV,SYS,MYUJV
```

To delete the exit routine for the specified SMF exit and subsystem, type

```
SMF DELETE,UJV,SYS,MYUJV
```

**Note**
Exits must be disabled before they can be deleted.

### Usage notes

- Exit routines are disabled when loaded. The exit routine will not be called by the SMF exit until it has been enabled.

- The exit routine is not actually loaded into storage until you use the ENABLE command. However, load processing validates the exit routine.

- For load requests, the data set name is required. However, the data set name is retained for the life of the MainView session and, therefore, need not be reentered on subsequent LOAD requests.

- The SMF exit (exit) calls the exit routine. You can load and enable multiple exit routines for the same exit. You can also load and enable the same exit routine for multiple SMF exits.

- Although multiple exit routines can be loaded and enabled for a single exit, each exit routine must have a unique name. For this reason, BMC Software recommends that you do not give your exit routine the same name as that of a supplied IBM exit routine. For example, use MYUJV instead of IEFUJV.

- An SMF exit is uniquely identified by the combination of the SMF exit name and the subsystem name. For example, SMF exit UJV and subsystem STC combine to identify an exit. SMF exit UJV and subsystem SYS combine to identify a different exit.

- For STATUS requests, you can specify any combination of parameters to limit the display. If you do not specify any parameters, all exit routines will be displayed.

---

**SMSLIST**

The SMSLIST (SMS) service displays information for IBM Storage Management System (SMS) managed volumes.
The SMSLIST service does not presently support the extended area of Extended Addressability Volume (EAV) devices.

The SMSLIST (SMS) service displays the following information for IBM Storage Management System (SMS) managed volumes:

- storage group names
- volume serial numbers
- percentage free space, largest free extent, and volume capacity by volume
- status

**Syntax**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>This keyword displays each storage group name followed by a list of volumes managed by that storage.</td>
</tr>
<tr>
<td>STATUS</td>
<td>This keyword has been retained for compatibility and is an alias for SGNAME.</td>
</tr>
<tr>
<td>SGNAME</td>
<td>The data is sorted by the storage group name in ascending sequence.</td>
</tr>
<tr>
<td>VOLUME</td>
<td>The data is sorted by volume serial number in ascending sequence.</td>
</tr>
<tr>
<td>FREE</td>
<td>The data is sorted by percentage of free space in descending sequence.</td>
</tr>
</tbody>
</table>

First positional parameter
### Keyword | Description
--- | ---
LXTNT | The data is sorted by the largest free extent in descending sequence.
TOTCAP | The data is sorted by total volume capacity in descending sequence.
MVS | The MVS status data is sorted in ascending sequence.
SMS | The SMS status data is sorted in ascending sequence.
FLAG | The data in the Error Flag column is sorted in descending sequence.

**Note**
All keywords can be abbreviated to the first three characters of the word.

### Second positional parameter

| Parameter | Description |
--- | ---
name | When specified, the second parameter limits the display. For LIST, only storage groups that match the value are displayed. For all other keywords, volumes with a storage group name or volume serial number that match are displayed.
An asterisk can be specified at the end of the value to indicate that the data is to be displayed when the characters preceding the asterisk match.

### Example 1

To display a list of the volumes that are controlled by storage groups that begin with SG, type

**SMS LIST SG***

| AMT1S1I Storage Group SGENG Volumes
| AMT1S2I SMSA13 SMSA12 SMSA01 SMSA04 SMSA03 SMSA06 SMSA05 SMSA08 SMSA11 SMSA07
| AMT1S2I SMSA10 SMSA16 SMSA15 SMSA14 SMSA09 SMSA02 SMSA00 SMSA18 SMSA17
| AMT1S0I AMT1S1I Storage Group SGSCOMP Volumes
| AMT1S2I SVC005 SVC004 SVC003 SVC002 SVC001 SVC006
| AMT1S0I AMT1S1I Storage Group SGSYS Volumes
| AMT1S2I SMS016 SMS015 SMS014 SMS012 SMS013 SMS011
| AMT1S0I AMT1S1I Storage Group SGTST Volumes
| AMT1S2I HFS001 HFS005 HFS000 SMS25D SMS25F SMS25A SMS25B HFS009 HFS008
| AMT1S2I SMS004 HFS003 SMS011 SMS009 SMS003 SMS008 SMS001 SMS007 HFS013 SMS25E
| AMT1S2I SMS006 SMS002 HFS010 HFS008 HFS019 HFS012 HFS018 HFS006 HFS017 SMS005
| AMT1S2I SMS25C HFS016 HFS002 HFS015 HFS014 HFS007 SMS25B
| AMT1S0I

### Example 2

To display information for all SMS-managed volumes that are sorted by descending percentage of free space with a VOLSER beginning with HFS, type

**SMS FREE HFS***

| AMTSSF2 Name | ID | MVS | SMS | %Free | Lxtnt | TotCap | Flag |
| AMTSSF1 -- SG -- Volume --- Status --- -------- Space -------- Error |

---

232  *MainView SYSPROG Services User Guide and Reference*
The SOFTFRR (SOF) service reads and summarizes software records from the LOGSTREAM or a LOGREC data set.

It also

- displays information about software errors
- displays software records for a specific module name

The SOFTFRR service displays the newest records first and the oldest records last. When records are obtained from the logstream, records older than seven days will not be selected.

**Syntax**

```
SOFTFRR
```

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modname</td>
<td>when specified, only records containing the specified <code>modname</code> in the recording parameters are selected</td>
</tr>
<tr>
<td>N/A</td>
<td>when specified, indicates that only records with a blank module name in the recording parameters are selected</td>
</tr>
<tr>
<td>*</td>
<td>when specified, indicates that all records are selected When specified without MAP, abstract data is displayed for each record, rather than a summary report.</td>
</tr>
<tr>
<td>MAP</td>
<td>indicates that all data in the selected software LOGREC records is displayed This data includes an English translation of all flags and the data in the Variable Recording Area, if any.</td>
</tr>
<tr>
<td>seqNum</td>
<td>when specified, only records containing the specified sequence number (error identification number) are selected</td>
</tr>
<tr>
<td>system</td>
<td>when specified, only the records for the specified system are selected The <code>system</code> operand is only supported when LOGREC is recording to the logstream.</td>
</tr>
<tr>
<td>yy.ddd</td>
<td>is the Julian date When specified, only records created on or after the specified Julian date are selected.</td>
</tr>
<tr>
<td>LOGSTREAM</td>
<td>when specified as the fourth parameter, indicates that records are obtained from the logstream, even if records are not currently being recorded in the logstream To display records from the current recording media, omit the fourth parameter.</td>
</tr>
<tr>
<td>dsname</td>
<td>is the name of the data set assumed to be a LOGREC data set when a value other than LOGSTREAM is specified as the fourth parameter The records are obtained from this data set even if it is not currently being used for recording errors. To display records from the current recording media, omit the fourth parameter.</td>
</tr>
</tbody>
</table>

**Note**

If you type **SOFTFRR** without operands, the service summarizes all software records in the LOGREC data set.

**Example 1**

To summarize all software records, type

**SOFTFRR**

```
AMTL3NI Processing logstream SYSPLEX.LOGREC.ALLRECS
AMTL31I  67 Records read;  17 Records accepted
AMTL39I  N/A  00003 IXGINLPA 00003 1GC00131 00001
AMTL39I  IKJEFLC  00003 ISLOAD  00001  NUCLEUS 00001
AMTL39I  BBM33  00001 XDCCALL  00001  IMF  00001
AMTL39I  BBI  00002  [1]  [2]
```

Legend:
1 lists all modnames for which there is data
   N/A (not available) means that the module identifier consists of binary zeros.

2 shows the number of records for each modname

**Example 2**

To display all of the data for records containing the modname BBM33, type

SOFT,BBM33,MAP

**Note**

This record was obtained from the logstream.

**Legend:**

1 displays the name of the system where data has been obtained

2 displays the error level PSWs and registers
   The six bytes of data preceding and following the PSW address at the time of the
   abend are also shown.

3 displays the RB level PSW and registers

4 translates the status flags
5 displays variable recording area (VRA) data, if any

**Example 3**

To display the modname and map for records that have data coming from the LOGREC, type:

```
SOFT,IKJEFT01,MAP
```

**AMTL30I Software Logrec Report (Vol=ES430M; Dsn=SYS1.LOGREC)**

- **AMTL32I ErrorID: CPUID: 0000; SeqNo: 937; Time: 14.26.53.3; ASID: 008A**
- **AMTL33I Abend S13E on 04/23/97 at 14.26.53.68 -- IKJEFT01.IKJEFTSC.IKJEFT05**
- **AMTL34I Err-PSW: 078C0000 80077A30; ILC 2; INTC 0001; Trans: 00000000 [1]**
- **AMTL35I Data at PSW-6 4100,0001,0A01,D503,401C,C4E8**
- **AMTL35I EFLGS: System issued SVC 13            |Enabled RB in control**
- **AMTL35I   :Cleanup only - Retry not allowed Abnd due to ancestor task error**
- **AMTL35I   :ErrorID information available**
- **AMTL36I Err-Regs  0-3  00000001 11A07F9B 00000000 00050A38**
- **AMTL36I Err-Regs  4-7  11A07F7C 11A323E8 00000000 000C68E0**
- **AMTL36I Err-Regs  8-11 00050A3B 00107F68 00077B84 0010EF28**
- **AMTL36I Err-Regs 12-15 80077740 000CB330 00007B7C 807B8658**
- **AMTL37I RB-PSW: 071C1000 810FF542; ILC 2; INTC 000D [2]**
- **AMTL38I RB-Regs  0-3  00000001 FF84C804 83F75728 00000000 000C86E0**
- **AMTL38I RB-Regs  4-7  007B741B 00000000 000B32F0 007FE030**
- **AMTL38I RB-Regs  8-11 00000000 007B32F0 007B32F0 007B32F0 007B76C8**
- **AMTL39I RB-Regs 12-15 007B32F0 007B32F0 007B32F0 007B32F0 007B32F0**
- **AMTL3D I Recovery routine percolated error [3]**
- **AMTL36I RFLGS: LOGREC recording requested**

**Note**

This record was obtained from SYS1.LOGREC.

**Legend:**

1. displays the error level PSWs and registers
   
The six bytes of data preceding and following the PSW address at the time of the
   abend are also shown.

2. displays the RB level PSW and registers

3. translates the status flags

4. displays variable recording area (VRA) data, if any

**Example 4**

To display LOGREC entries for a specific system, the system operand must be
specified where LOGREC is recording to a logstream.
The SPACE (SPA) service locates DASD devices based on the specified selection criteria and displays space-related information for the selected devices. You can also control the sort order of the displayed information.

**Note**
As of Release 3.6, the SPACE service supports the cylinder-managed area of EAV volumes and includes other new features. The amount of storage space is displayed in bytes, rather than in tracks and cylinders.

The following information is displayed:

- device address
- VOLSER
- mount attribute
- EAV volume flag

You can also specify that you want information for the track-managed area (TRACK), cylinder-managed area (CYLINDER), or the two areas combined (COMBINED). The default is TRACK. The following information is displayed for the specified area as follows:

- total area size in bytes
- free space in bytes
- largest extent in bytes
- percentage of free space
- fragmentation index

Other filters are available to limit the displayed information.
Syntax

Operands
The first operand must be one of the following choices:

- device number
- esoteric (or generic) name
- volume serial number
- partial volume serial number

A partial volume serial number is specified by appending an asterisk (*) to the desired characters. For example, ABC* indicates that all devices with a volume serial number beginning with ABC are to be selected. Specifying only an asterisk indicates that all DASD volumes are desired and is equivalent to using the esoteric name SYSALLDA. The default is the generic name SYSDA.

The following operands are not positional and can be specified in any sequence. The last operand controls the sort order.

Subsequent operands allows you to limit the display to the desired information and to control how that information is sorted. You can specify as many operands as needed to obtain the desired result. Several keywords control the sort order, in which case the last keyword specified prevails. All keywords can be abbreviated to their first three characters. Keywords that accept an operator permit you to reject volumes that are not within the specified range. For example, you could specify LARGEST>2T to restrict the display to volumes that have at least two terabytes of contiguous free space, or you could specify SPACE * CYLINDER PCT<10 to locate all EAV volumes with free space of 10% or less.

The syntax diagram has two additional columns of logically grouped options as follows:
TRACK, CYLINDER, and COMBINED are mutually exclusive options that select the management area to be processed.

— TRACK (the default) indicates that you want data for the traditional track-managed area that is applicable to all volumes.

— CYLINDER indicates that you want only information for the cylinder-managed area on Extended Addressability Volumes (EAV) devices.

— COMBINED indicates that you want the information for both areas combined.

ALLATR, PUBLIC, PRIVATE, and STORAGE filters (limits) the display to volumes with the requested mount attributes. The default is ALLATR, which indicates that you want all three. You can specify more than one value. For example, specifying PUBLIC, STORAGE causes PRIVATE volumes to be omitted.

The variable, exp, is an expression that provides additional filtering options and controls how the data is sorted. The following two forms for exp are provided:

\[
\text{label} \\
\text{label operator value}
\]

- \text{label} is one of the following column header labels: EV, TOTAL, FREE, LARGEST, PCT, and FRAG.
- \text{operator} is one of the following operands: <, =, or >.
- \text{value} is the desired limit value.

The variable \text{value} can be N or Y for keyword EAV. You can choose

- \text{EAV=N} to reject all EAV volumes
- \text{EAV=Y} to reject all non-EAV volume
- \text{EAV} to sort by the EAV column

For all other keywords value is any numeric value and, to simplify working with large values, you can append the numeric value with one of the following multipliers: K, M, G, T, or P, each of which increases by 1,000.

Table 6 on page 240 lists all of the supported keywords and indicates which ones affect sort order, the direction of the sort, and if you can append operator/value pairs to the keyword.
Table 6: SPACE service keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Sort order</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLATR</td>
<td>select volumes regardless of their mount attribute (default)</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>ATR</td>
<td>sort by mount attribute (PUBLIC, PRIVATE, STORAGE)</td>
<td>ascending</td>
<td></td>
</tr>
<tr>
<td>COMBINE</td>
<td>combined track- and cylinder-managed data is desired</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>CYLINDER</td>
<td>indicates that cylinder-managed area data is desired</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>EAV</td>
<td>sort by EAV field or limit to EAV=N or EAV=Y</td>
<td>ascending</td>
<td>=</td>
</tr>
<tr>
<td>FRAG</td>
<td>sort by Fragmentation Index value and optionally limit to volumes with less than or more than the specified Fragmentation Index value</td>
<td>descending</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>FREE</td>
<td>sort by largest contiguous free extent and optionally limit to volumes with less than or more than the specified contiguous free space</td>
<td>descending</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>LARGEST</td>
<td>sort by largest contiguous free extent and optionally limit to volumes with less than or more than the specified contiguous free space</td>
<td>descending</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>PCT</td>
<td>sort by the percentage of free space and optionally limit to volumes with less than or more than the specified percentage of free space</td>
<td>descending</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>select volumes with a mount attribute of PRIVATE*</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>PUBLIC</td>
<td>select volumes with a mount attribute of PUBLIC*</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>STORAGE</td>
<td>select volumes with a mount attribute of STORAGE*</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>TOTAL</td>
<td>sort by the total size of the referenced area and optionally limit to volumes with less than or more than the specified value</td>
<td>descending</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>TRACK</td>
<td>indicates that track-managed area data is desired</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>UNIT</td>
<td>sort by device address</td>
<td>ascending</td>
<td>NA</td>
</tr>
<tr>
<td>VOLUME</td>
<td>sort by volume serial number</td>
<td>ascending</td>
<td>NA</td>
</tr>
</tbody>
</table>
Example 1

To display track-managed area information for all devices with a volume serial number that begins with BAB36 and sort the information by the percentage of free space in descending sequence (the default), type

```
SPACE BAB36*
```

```
10 devices selected
```

<table>
<thead>
<tr>
<th>Unit</th>
<th>Volume</th>
<th>Atr</th>
<th>EAV</th>
<th>Bytes</th>
<th>Free</th>
<th>Extent</th>
<th>Pct</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>85C9</td>
<td>BAB369</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>1,030M</td>
<td>356M</td>
<td>34%</td>
<td>273</td>
</tr>
<tr>
<td>85C7</td>
<td>BAB367</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>857M</td>
<td>306M</td>
<td>28%</td>
<td>186</td>
</tr>
<tr>
<td>8565</td>
<td>BAB365</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>742M</td>
<td>352M</td>
<td>25%</td>
<td>124</td>
</tr>
<tr>
<td>8560</td>
<td>BAB360</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>603M</td>
<td>283M</td>
<td>20%</td>
<td>224</td>
</tr>
<tr>
<td>8564</td>
<td>BAB364</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>586M</td>
<td>446M</td>
<td>19%</td>
<td>87</td>
</tr>
<tr>
<td>8561</td>
<td>BAB361</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>587M</td>
<td>342M</td>
<td>19%</td>
<td>92</td>
</tr>
<tr>
<td>8562</td>
<td>BAB362</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>521M</td>
<td>376M</td>
<td>17%</td>
<td>103</td>
</tr>
<tr>
<td>8568</td>
<td>BAB368</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>512M</td>
<td>390M</td>
<td>17%</td>
<td>140</td>
</tr>
<tr>
<td>8561</td>
<td>BAB361</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>472M</td>
<td>307M</td>
<td>15%</td>
<td>91</td>
</tr>
</tbody>
</table>

Example 2

To locate all volumes with a volume serial number beginning with BAB, a mount attribute of PRIVATE, and at least one gigabyte of contiguous free space sorted by descending contiguous free space, type

```
SPA,BAB*,PRI,LARGEST>1G
```

```
8 devices selected
```

<table>
<thead>
<tr>
<th>Unit</th>
<th>Volume</th>
<th>Atr</th>
<th>EAV</th>
<th>Bytes</th>
<th>Free</th>
<th>Extent</th>
<th>Pct</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>839D</td>
<td>BAB908</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>2,124M</td>
<td>1,921M</td>
<td>72%</td>
<td>44</td>
</tr>
<tr>
<td>8354</td>
<td>BAB907</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>2,333M</td>
<td>1,495M</td>
<td>79%</td>
<td>134</td>
</tr>
<tr>
<td>831A</td>
<td>BAB201</td>
<td>PRV</td>
<td>N</td>
<td>1,910M</td>
<td>1,486M</td>
<td>1,461M</td>
<td>77%</td>
<td>8</td>
</tr>
<tr>
<td>8319</td>
<td>BAB200</td>
<td>PRV</td>
<td>N</td>
<td>1,910M</td>
<td>1,330M</td>
<td>1,330M</td>
<td>69%</td>
<td>0</td>
</tr>
<tr>
<td>85C4</td>
<td>BAB904</td>
<td>PRV</td>
<td>Y</td>
<td>2,944M</td>
<td>2,112M</td>
<td>1,262M</td>
<td>71%</td>
<td>119</td>
</tr>
<tr>
<td>876E</td>
<td>BAB906</td>
<td>PRV</td>
<td>N</td>
<td>2,944M</td>
<td>2,079M</td>
<td>1,201M</td>
<td>70%</td>
<td>116</td>
</tr>
<tr>
<td>AC0F</td>
<td>BAB914</td>
<td>PRV</td>
<td>N</td>
<td>8,833M</td>
<td>1,397M</td>
<td>1,180M</td>
<td>15%</td>
<td>57</td>
</tr>
<tr>
<td>AC0C</td>
<td>BAB911</td>
<td>PRV</td>
<td>N</td>
<td>8,833M</td>
<td>1,859M</td>
<td>1,097M</td>
<td>20%</td>
<td>153</td>
</tr>
</tbody>
</table>

Example 3

To locate all Extended Addressability Volume (EAV) devices and display information for the combined track and cylinder-managed areas, type

```
SPA*,EAV=Y,COMBINED
```

```
4 devices selected
```

<table>
<thead>
<tr>
<th>Unit</th>
<th>Volume</th>
<th>Atr</th>
<th>EAV</th>
<th>Bytes</th>
<th>Free</th>
<th>Extent</th>
<th>Pct</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>2401</td>
<td>EAVNS1</td>
<td>PRV</td>
<td>Y</td>
<td>60,849M</td>
<td>20,199T</td>
<td>30,116M</td>
<td>49%</td>
<td>3</td>
</tr>
<tr>
<td>2200</td>
<td>EVANLA</td>
<td>PRV</td>
<td>Y</td>
<td>232G</td>
<td>12,372T</td>
<td>44,887M</td>
<td>21%</td>
<td>34</td>
</tr>
<tr>
<td>2400</td>
<td>EAVNL1</td>
<td>PRV</td>
<td>Y</td>
<td>232G</td>
<td>32,023T</td>
<td>118G</td>
<td>72%</td>
<td>73</td>
</tr>
<tr>
<td>2201</td>
<td>EVANSA</td>
<td>PRV</td>
<td>Y</td>
<td>60,849M</td>
<td>17,421T</td>
<td>6,406M</td>
<td>81%</td>
<td>188</td>
</tr>
</tbody>
</table>
The SRM service displays system resources manager values and data elements used for job and system management.

The addresses of SRM control blocks are also provided so you can modify control fields by using the ZAP service. This service also determines the current status of key SRM control values.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>MAP</td>
<td>displays the addresses of SRM control blocks</td>
</tr>
</tbody>
</table>

**Example 1**

To display system resource information, type

```
SRM
```

AMTS301 System is in goal mode
AMTS311 Address Spaces : In=42 O/Rdy=0 O/Nrdy=29 Logwt=63
AMTS321 Resource Status : CPU=8% High UIC=1800
AMTS361 Paging Rate/Sec: Demand=0 Total=45 Delay Time=16,384
AMTS391 Think-time(Sec): Cur=0.0 Min=0.0 Max=5.0

**Example 2**

To display the addresses of SRM control blocks as well, type

```
SRM MAP
```

AMTS301 System is in goal mode
AMTS311 Address Spaces : In=40 O/Rdy=0 O/Nrdy=29 Logwt=65
AMTS321 Resource Status : CPU=2% High UIC=1900
AMTS361 Paging Rate/Sec: Demand=0 Total=0 Delay Time=22,016
AMTS391 Think-time(Sec): Cur=0.0 Min=0.0 Max=5.0

AMTS331 RMCT 0188D420 RMCA 0188DC90 RMPT 0188DC08 CCT 0188D708
Legend:

1. number of address spaces currently swapped in memory
2. number of address spaces swapped out of memory but ready
3. number of address spaces out of memory and not ready
4. current SRM-weighted percentage of CPU busy
5. highest unreferenced interval count

Example 3

To display system resource information for address space BMVDWP5, type:

SRM,BMVDWP5

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS71I</td>
<td>TSU09423 BMVDWP5</td>
<td>IKJDWP2 Service Class=TSNORM</td>
<td></td>
</tr>
<tr>
<td>AMTS72I</td>
<td>Transaction data: Interval 00.00.07 [5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS73I</td>
<td>Last Swap : Interval 00.00.07 Reason= &lt; IN MEMORY &gt;[6]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS74I</td>
<td>Swapping Profile: Count 1 [7]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS75I</td>
<td>Service Data : CPU 1,264 [8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS76I</td>
<td>: SRB 29 [8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS77I</td>
<td>: I/O 190 [8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS78I</td>
<td>: MSO 0 [8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS79I</td>
<td>: Total 1,483 [9]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1. Job Identifier
2. Job name
3. Step name
4. Service Class
5. Duration of this particular transaction
6. Time of the last swap and the reason for the swap
7. Swap count during the transaction
8. CPU, SRB, I/O, and MSO service units used by the transaction
9. Total service units used by the transaction
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM LOW</td>
<td>swapped out due to auxiliary storage shortage</td>
</tr>
<tr>
<td>ENQ EX</td>
<td>swapped out due to enqueue exchange</td>
</tr>
<tr>
<td>IN MEMORY</td>
<td>swapped in</td>
</tr>
<tr>
<td>LONG WT</td>
<td>swapped out due to long wait</td>
</tr>
<tr>
<td>MSO WAIT</td>
<td>swapped out due to MSO-detected long wait</td>
</tr>
<tr>
<td>REC EX</td>
<td>swapped out due to recommendation values</td>
</tr>
<tr>
<td>REQUEST</td>
<td>swapped out due to a request swap</td>
</tr>
<tr>
<td>RSM LOW</td>
<td>swapped out due to a real-storage shortage</td>
</tr>
<tr>
<td>TERM IN</td>
<td>swapped out due to terminal input wait</td>
</tr>
<tr>
<td>TERM OUT</td>
<td>swapped out due to terminal output wait</td>
</tr>
<tr>
<td>UNILTRL</td>
<td>swapped out due to unilateral swap-out</td>
</tr>
</tbody>
</table>

10 pages swapped at last swap-out, and the working-set size at swap-in

11 service units for this transaction; values are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>CPU service units</td>
</tr>
<tr>
<td>I/O</td>
<td>I/O service units</td>
</tr>
<tr>
<td>MSO</td>
<td>main storage occupancy service units</td>
</tr>
<tr>
<td>TOTAL</td>
<td>total service units for this transaction</td>
</tr>
</tbody>
</table>

Note
The appearance of *** means that these values are temporarily invalid because they are being updated by SRM. Invoke the SRM service again to display updated values.

SSVT

The SSVT (SSV) service maps information from the Subsystem Communications Vector Table (SSCVT).

This service can also list information for a single subsystem and dynamically add a subsystem without having to re-IPL. This service also identifies the subsystems used and locates their respective Subsystem Vector Table (SSVT) control blocks.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subsystemID</td>
<td>is the name of the subsystem that you want to list or add</td>
</tr>
<tr>
<td>LIST</td>
<td>lists information about the specified subsystem from the Subsystem Communications Vector Table (SSCVT)</td>
</tr>
<tr>
<td>ADD</td>
<td>dynamically adds the specified subsystem to the system</td>
</tr>
<tr>
<td></td>
<td>The subsystem remains on the system until the next IPL. To permanently add the subsystem, you must modify SYS1.PARMLIB(IEFSSN00).</td>
</tr>
</tbody>
</table>

**Note**

If you type SSVT without parameters, SSCVT information from all subsystems is listed.

Example

To map information from the SSCVT, type

```
SSVT
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTB611</td>
<td>SSCVT=005D26EB NAME=JES2 (D1C5E2F2) SSVT=005D5450 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=005D26C0 NAME=MSTR (D4E2E3D9) SSVT=00950EE8 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=005D2010 NAME=CICS (C3C9C3E2) SSVT=00000000 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=00950908 NAME=JES1 (D1C5E2F1) SSVT=00000000 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=009508E0 NAME=OCCF (D6C3C3C6) SSVT=00000000 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=009508B8 NAME=RES. (D9C5E240) SSVT=00000000 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=009490F8 NAME=SIMS (E2C9D4E2) SSVT=00000000 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=00631698 NAME=VX.. (E5E7000E) SSVT=00610EC8 SUSE=00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=006621C8 NAME=IMF1 (C9D4C6F1) SSVT=006621F0 SUSE=0093A3D0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTB611</td>
<td>SSCVT=006648C8 NAME=RB20 (D9C2F2FO) SSVT=00664938 SUSE=008ECBDO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. address of the SSCVT for this subsystem
2. subsystem name
3. subsystem name in hexadecimal format
4. address of the SSCVT for this subsystem; zero if the subsystem is not active
The status of the SSCVTSUSE field for this subsystem

**STATUS**

The STATUS (STA) service monitors and displays the current status of active address spaces in the system.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>displays the status of all address spaces swapped in memory (default)</td>
</tr>
<tr>
<td>ALL</td>
<td>displays the status of all address spaces</td>
</tr>
<tr>
<td>TSO</td>
<td>displays the status of all TSO users</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>SHORT</td>
<td>shortens the display to accommodate console screen width</td>
</tr>
</tbody>
</table>

**Example**

To display the status of all active address spaces swapped in memory, type

```
STATUS
```

Legend:
1. address space name
2. address space ID (ASID)
3. status indicator #1; possible values are as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSW</td>
<td>nonswappable</td>
</tr>
<tr>
<td>LS</td>
<td>logically swapped</td>
</tr>
<tr>
<td>PVL</td>
<td>privileged</td>
</tr>
<tr>
<td>OUT</td>
<td>swapped out of memory</td>
</tr>
<tr>
<td>GOI</td>
<td>going in</td>
</tr>
<tr>
<td>GOB</td>
<td>going between states</td>
</tr>
<tr>
<td>ENQ</td>
<td>enqueue privileged</td>
</tr>
<tr>
<td>IN</td>
<td>in memory</td>
</tr>
<tr>
<td>GOO</td>
<td>going out of memory</td>
</tr>
</tbody>
</table>

4. real frames currently in use by this address space
5. current working set in pages
   If a job is nonswappable, the working set is displayed as ***. The current working set is the number of page frames swapped at the last swap action. This value is also the number of page frames necessary to swap in an address space.

6. CPU time used by the current step in this address space, in seconds and hundredths

STOKEN

The STOKEN (STO) service displays the address space token (STOKEN) for the specified address space in hexadecimal.

**Syntax**

```
STOKEN asl
```

The parameter is defined as follows:
### SUBMIT

The SUBMIT (SUB) service submits a job from the specified data set to the job entry subsystem.

This service enables you to submit JCL for routine operations jobs.

Because the JCL contains a JOB statement, a valid user account code can be used for accounting purposes.

#### Syntax

```
SUBMIT member dsname
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>member</code></td>
<td>is a member in the BBPARM library data set</td>
</tr>
<tr>
<td><code>dsname</code></td>
<td>is the name of the data set containing the member to be submitted</td>
</tr>
</tbody>
</table>

If you do not specify a data set name, the data set last specified on a SUBMIT command is used. Therefore, if you are submitting multiple members from the same data set, you only need to specify the data set name on the first SUBMIT command. If you have not specified a data set name and DDNAME BBPARM was preallocated, that data set will be used.

#### Examples

To submit BBPARM library member `backup` to the job entry subsystem, type
SUBMIT BACKUP

To submit the jobs in member backup in data set MY.PARMLIB, type

SUBMIT BACKUP,MY.PARMLIB

If this data set name was previously entered, the previous data set MY.PARMLIB will be used. If a data set was not previously specified and DDNAME BBPARM was preallocated, that data set will be used.

To submit the jobs in member backup2, type

SUBMIT BACKUP2

SVCFIND

The SVCFIND (SVC) service scans the SVC table, locating and displaying information about SVCs.

It also

■ dumps the first 64 bytes of the SVC module from main storage
■ helps you plan the installation of new SVCs

For information about loading SVCs into main storage, see “LOADLPA” on page 143 on page 182.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type#</td>
<td>is a one-digit SVC type number (1, 2, 3, 4, or 6) SVCFIND displays unused SVC table entries that equal the type number.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **starting#** | is a one- to three-digit SVC number  
The display begins with this number. The default is 200. |
| **ending#** | is a one- to three-digit SVC number  
The display ends with this number. The default is 255. |
| **LIST** | displays all SVC numbers in the specified starting number and ending number range |
| **DUMP** | displays the first 64 bytes of the specified svc# from main storage |
| **svc#** | is a one- to three-digit SVC number  
This number cannot exceed 255. The default is 0. |

**Note**

If no operands are specified, SVCFIND displays all unused SVC table entries, regardless of type number. This option is the default.

**Examples**

To display SVC table entries for unused type 2 SVC members between 240 and 245, type

**SVCFIND 2 240 245**

AMTY11I UNUSED SVCTABLE ENTRIES 240-245 TYPE=2

AMTY13I SVC# TYPE EPA APF NP ESR ASF LOCKS TABLE-ADR FLAGS
AMTY14I 240 2 -UNUSED- N N N Y NONE 00FDB4A0 81000000
AMTY14I 243 2 -UNUSED- N N N Y NONE 00FDB4B0 81000000

To display all SVC types, used and unused, between 199 and 201, type

**SVCFIND LIST 199 201**

AMTY11I SVCTABLE ENTRIES 199-201

AMTY13I SVC# TYPE EPA APF NP ESR ASF LOCKS TABLE-ADR FLAGS
AMTY14I 199 2 -UNUSED- N N N Y NONE 00FDB358 81000000
AMTY14I 200 2 -UNUSED- N N N Y NONE 00FDB360 81000000
AMTY14I 201 2 00E1AC10 Y N N Y NONE 00FDB368 C8000000

To dump the first 64 bytes of storage for SVC 235, type

**SVCFIND DUMP 235**

AMTY15I SVC 235 DISPLAY

AMTY16I 813BA1C0 47F0601A 14C9C7C3 F0F0F340 F0F3F1F9 *..-..16C003.019*
AMTY16I 0138A1D0 F540C8C2 C2F7F7F0 F0004120 6FF189F *5.HBB7790...?...*
AMTY16I 0138A1E0 18C11830 41F00000 58002889 5810706C *...A....0.....1...%*
AMTY16I 0138A1F0 58101070 5810101C 1881181C 1803500D *.........a....&.*

**Legend:**

1 bit settings in the SVCTABLE entry

Refer to the IBM Publication, *z/OS MVS Data Areas Vol 5*, for a description of the SVCTABLE data area.
highest lock required by the SVC first-level interrupt handler; the lock is obtained from the \texttt{SVCLOCKS} field of the \texttt{SVCTABLE}

address of the \texttt{SVCTABLE} entry for the appropriate SVC

hexadecimal display of the \texttt{SVCATTR} and \texttt{SVCLOCKS} fields from the \texttt{SVCTABLE} entry

starting address of the storage being dumped, and the hexadecimal and EBCDIC representation of that data

text representation of the dump

\section*{SYSDUMP}

The SYSDUMP (SYS) service displays the title of the dump, as well as the date and time that the dump was taken for each active SYS1.DUMPxx data set.

This service also determines if SVC dumps should be retained or deleted, which helps prevent the loss of important dumps due to all dump data sets being full. (See “CLEAR” on page 72 for information about clearing dump data sets.)

\subsection*{Syntax}

\begin{verbatim}
SYSDUMP
\end{verbatim}

\subsection*{Example}

To display information about each active dump data set, type

\begin{verbatim}
SYSDUMP
\end{verbatim}

\begin{table}[h]
\begin{tabular}{ll}
\textbf{AMTDU0I} & \texttt{SYS1.DUMP01} WAS FILLED AT 10:21 ON MAR 14, 2003 \textsuperscript{[2]} \\
\textbf{AMTDU1I} & \texttt{SOURCE=SVCDUMP} \textsuperscript{[3]} \\
\textbf{AMTDU2I} & \texttt{TITLE=ERROR IN TSO} \textsuperscript{[4]}
\end{tabular}
\end{table}

\textbf{Legend:}

1 full-dump data set

2 time and date of dump

3 system component that took the dump
The TCB service displays information for the various tasks (TCBs) within an address space or a specific task (TCB).

When all tasks are displayed, a symbolic name is generated for each TCB. When used with another TCB service command, the service will display the detailed information for the task that assigns symbols to most of the displayed fields. These symbols may be used with several other services such as CTCB, DUMP, POST, WHERE, and ZAP.

**Syntax**

```
TCBasi@TCBn MAP ALL
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>Address space instance; see “Identifying address spaces” on page 47 for more information. When an asi is provided, summary information for all TCBs is displayed, except for the first two TCBs. See the ALL parameter for more information.</td>
</tr>
<tr>
<td>@TCBn</td>
<td>Is a symbolic name generated by a previous invocation of the TCB service. Detailed information for the specified TCB is displayed and symbolic names are assigned to the values in most of the displayed fields. See the list of generated symbols for details. One symbol is generated for each TCB within the target address space, and the symbols are numbered consecutively beginning with 001. The symbols are deleted and reassigned each time the TCB service is invoked specifying any value other than @TCBn as the first parameter.</td>
</tr>
<tr>
<td>MAP</td>
<td>Ignored when @TCBn is specified as the first parameter. When an asi is specified as the first parameter, indicates that detailed information for all of the TCBs (except the first two, see the ALL parameter) is to be displayed. However, symbolic names are not assigned to the various fields.</td>
</tr>
<tr>
<td>ALL</td>
<td>Ignored when @TCBn is specified as the first parameter. When an asi is specified as the first parameter, indicates that information for the first two TCBs is to also be displayed.</td>
</tr>
</tbody>
</table>
Example 1

To display summary information for all of the tasks (except the first two) within the address space with jobname VAM3R7, type

TCB,VAM3R7

Example 2

To display detailed information for the ninth TCB displayed in example 1 above, which has a symbolic name of @TCB9, type

TCB,@TCB9
In this example, the high-order word of the 16 general purpose registers is displayed because at least one of those words is non-zero. The access registers were not displayed because they all contained zero.

**Example 3**

After displaying detailed information for the TCB, you might want to use other services to display other information. For example, you can use the DUMP service to display the contents of storage referenced by the task’s registers; you can use the WHERE service to determine the subpool and other information. To display the contents of storage at R12 (which refers to the 64-bit address in general register 12), type

DUMP,R12

```
AMTC1ZI 00000000_2F32C600 KEY= 0 VAM3R7 (ASID 005A)
AMTC1ZI 2F32C600 A7F40044 58C1E2E3 58E2CAE5 D95EF0F5 *x4  AST$SDVR:05*
AMTC1ZI 2F32C610 61F0F261 F1F25EF1 F24BF4F4 5EC4C5E5 */02/12.12.44;DEV*
AMTC1ZI 2F32C620 60F34503 5E404040 40404040 5E404040 *_LVL: ;*
AMTC1ZI 2F32C630 40404040 5E5E5E5E 2F32C6A4 5E4DC35D *x4  Fu;(C)*
```

To display location-related information (such as the subpool, key, and load module name if applicable) and offset into the load module, type

WHERE,R12

```
AMTW4I 2F32C600 Found in Extended Private storage
AMTW5I Subpool 252 (Pageable), key 0, Owned by TCB 6FF3A0 (@TCB4)
AMTW6I At AST$FV3+3600 Load Adrs=2F329000, Len=00008000
```

**Example 4**

To display more detailed information about a TCB, type

TCB,@TCB5

```
AMTT71I Address Space: XTST6PAS.PAS6 Step: PAS6
AMTT71I @TCB5 TCB 6EE220 Prty 255 NDSP 0400 PK 40
AMTT71I 00000000 00000000 00000000 00000000 [10]
AMTT71I 00000000 00000000 00000000 00000000 [10]
AMTT71I 00000000 00000000 00000000 00000C05 [10]
AMTT76I The task is Waiting on ECB1 at 006C6CF8
AMTT76I The task is Waiting on ECB2 at 7F0E9E34
```

**Generated symbols**

In addition to generating a symbol for each TCB that refers to the address of the TCB it represents, the following symbols are generated whenever detailed information for a specific TCB is requested.
Table 7: TCB address symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBBASIC</td>
<td>The address of the basic section (RBBASIC) of the last (most recently created) RB for the task.</td>
</tr>
<tr>
<td>PSW</td>
<td>The address contained in the second word of the PSW in the last (most recently created) RB for the task.</td>
</tr>
<tr>
<td>R0 through R15</td>
<td>The 64-bit value for the last (most recently created) RB for the task.</td>
</tr>
<tr>
<td>ECBn</td>
<td>address of an ECB upon which the task is waiting. One symbol is generated for each ECB upon which the task is waiting.</td>
</tr>
</tbody>
</table>

TIME

The TIME (TIM) service displays date and time.

The date is displayed in both Gregorian and Julian formats. The Greenwich Mean Time (GMT) time display includes the offset to the local time.

Syntax

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMT</td>
<td>displays the Greenwich Mean Time specified by the operator at the time of IPL</td>
</tr>
</tbody>
</table>

Example 1

To display the current date and time, type

```
TIME
```

Legend:

1 day of the week
2 Gregorian date

3 Julian date

4 local time in hours, minutes, seconds, and hundredths of seconds

**Example 2**

To display the current date and GMT time, type

```
TIME,GMT
```

**Note**

You can type the TIME GMT command with or without a comma.

**Legend:**

1 day of the week

2 Gregorian date

3 Julian date

4 Greenwich Mean Time

5 local time offset from Greenwich Mean Time

**TIOT**

The TIOT (TIO) service displays all allocated devices for a specified address space and monitors the usage of data sets by address space.

**Syntax**

```
TIOT  asi
```

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
<tr>
<td>MAP</td>
<td>displays the data set names associated with each TIOT entry</td>
</tr>
<tr>
<td>ACT</td>
<td>displays the active TIOT entries that contain data sets with nonzero EXCP counts. The data set names are listed, as is the case with MAP.</td>
</tr>
<tr>
<td>ddname</td>
<td>is the name of a DD statement. Specifying this parameter displays the data set names associated with the specified DD name.</td>
</tr>
</tbody>
</table>

**Note**

Within a concatenation, if more than one of the data sets are allocated to a common volume, TIOT attributes all the EXCPs for the data sets sharing that volume to the first data set within the concatenation that is on the current volume. For data sets sharing a common volume within a concatenation, IBM updates the TCT I/O table only for the first data set in the concatenation that resides on that volume.

**Example 1**

To display all allocated devices for address space INVENTORY and display the data set name associated with each TIOT entry, type

```
TIOT INVENTORY MAP
```

| AMTE11I JOB 332 INVENTORY UPDATE1 PRTY 279 PAGE I/O’S 1044 |
|-------------|-----------------|-----------------|-----------------|-----------------|
| AMTE12I DD STIEPLIB UNIT 151 VOLUME SYS001 EXCP 604       |
| AMTE13I DSN SYS3.SYPROG LOAD                               |
| AMTE12I UNIT 250 VOLUME PERF01                             |
| AMTE13I DSN BB.CMC.CMC310 LOAD                             |
| AMTE12I UNIT 250 VOLUME PERF21 EXCP 129                    |
| AMTE13I DSN BB.CMC.CMC310 LINK                             |
| AMTE12I DD SYS00002 UNIT 254 VOLUME PUB001                 |
| AMTE13I DSN SYSCTLG.PUB001                                 |
| AMTE12I DD SYS00003 UNIT 254 VOLUME PUB002 EXCP 15         |
| AMTE13I DSN SYSCTLG.PUB002                                 |

**Legend:**

1. JES job ID
2. name of the specified address space
3. current stepname for the specified address space
4. internal dispatching priority for the address space
5. total page faults requiring a page I/O operation for this address space
6 DD name
7 device number
8 serial number of volume mounted on device
   Only the first volume for multi-volume data sets are listed. You can use the DSNAME service to display all of the volumes for a multi-volume data set.
9 EXCP count for each data set within each DD name
10 data set name associated with TIOT entry; issued only if you specify MAP, ACT, or ddname

Example 2

To limit output to TIOT entries that have data sets with nonzero EXCP counts, type

```
TIOT INVENTORY ACT
```

```
AMTE11I JOB 332 INVENTORY UPDATE1 PRTY 279 PAGE I/O'S 1044
AMTE12I DD STEPLIB UNIT 151 VOLUME SYS001 EXCP 604
AMTE13I DSN SYS3.SYSPROG.LOAD
AMTE12I UNIT 250 VOLUME PERF21 EXCP 129
AMTE13I DSN BB.CMC.CMC310.LINK
AMTE12I DD SYS00003 UNIT 254 VOLUME PUB002 EXCP 15
AMTE13I DSN SYSCTLG.PUB002
```

Example 3

To limit output to the data set names associated with the DD name STEPLIB, type

```
TIOT INVENTORY,STEPLIB
```

```
AMTE11I JOB 332 INVENTORY UPDATE1 PRTY 279 PAGE I/O'S 1044
AMTE12I DD STEPLIB UNIT 151 VOLUME SYS001 EXCP 604
AMTE13I DSN SYS3.SYSPROG.LOAD
AMTE12I UNIT 250 VOLUME PERF01
AMTE13I DSN BB.CMC.CMC310.LOAD
AMTE12I UNIT 250 VOLUME PERF21 EXCP 129
AMTE13I DSN BB.CMC.CMC310.LINK
```

TOD

The TOD service formats and displays an eight-byte time of day (TOD) clock value.

You can specify the values to be formatted as a hexadecimal value, or you can specify an expression, that when evaluated within the context of the specified address space, denotes the address of the TOD value to be formatted. Optionally, you can request that the value be converted from GMT to local time or from local time to GMT.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>expression</strong></td>
<td><em>expression</em> can be any valid expression as described in the section titled “Expressions” on page 48. It can also be an eight- or four-byte hexadecimal literal value enclosed within quotation marks and prefixed with an X.</td>
</tr>
<tr>
<td>LOCAL</td>
<td><em>(optional)</em> assumes that the value is in GMT and is to be converted to local time</td>
</tr>
<tr>
<td>GMT</td>
<td><em>(optional)</em> assumes that the value is in local time and is to be converted to GMT</td>
</tr>
</tbody>
</table>
| **asi** | *(optional)* denotes the address space that contains the TOD value addressed by the expression  
If omitted, the address space implied by the expression (if any) will be used. If no address space is expressly specified or implied, the user’s address space is used. |

**Note**
The *asi* variable and LOCAL | GMT can be specified in either the second or the third parameter position.

**Example 1**

Format and display the date and time the JES2 address space was initiated. Convert the time to LOCAL time.

**TOD,ASCB+130,JES2,LOCAL**

**AMTTD1I** 10:19:19.34 PM  
**AMTTD2I** Wednesday, May 12, 2010 Local time  
**AM2001A** SYSPROG

**Example 2**

Format and display the hexadecimal value X' C45B3A532692C383' in date and time format.

**tod,X'C45B3A532692C383'**

**AMTTD1I** 6:13:13.53 PM  
**AMTTD2I** Thursday, June 18, 2009 (GMT assumed)
Example 3

Convert to local time, format and display the hexadecimal value X'\text{C45B3A532692C383}' in date and time format.

\text{tod X'C45B3A532692C383',local}

TPIO

The TPIO (TPI) service displays all outstanding I/O, including TP devices.

This service also determines if TP lines are enabled (have outstanding I/O).

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{dvn}</td>
<td>is the device number for which outstanding I/O is to be displayed</td>
</tr>
<tr>
<td>\text{volser}</td>
<td>is the volume serial number for which outstanding I/O is to be displayed</td>
</tr>
</tbody>
</table>

Example 1

To display all outstanding I/O, type

\text{TPIO}

Legend:

1 address space name

2 device number

3 volume serial number of the device, if DASD or a tape device
4 IOQ address

5 code of the IOS driver, which can be as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISC</td>
<td>miscellaneous driver</td>
</tr>
<tr>
<td>EXCP</td>
<td>EXCP driver</td>
</tr>
<tr>
<td>VSAM</td>
<td>VSAM driver</td>
</tr>
<tr>
<td>VTAM</td>
<td>VTAM driver</td>
</tr>
<tr>
<td>TCAM</td>
<td>TCAM driver</td>
</tr>
<tr>
<td>OLTEP</td>
<td>OLTEP driver</td>
</tr>
<tr>
<td>PCIFETCH</td>
<td>program FETCH driver</td>
</tr>
<tr>
<td>JES3</td>
<td>JES3 subsystem</td>
</tr>
<tr>
<td>MSS/MSC</td>
<td>mass storage subsystem</td>
</tr>
<tr>
<td>IOSPURGE</td>
<td>internal IOS PURGE IOQ routine</td>
</tr>
<tr>
<td>VPSS</td>
<td>vector processing subsystem (3838 array processor)</td>
</tr>
<tr>
<td>CRYPTO</td>
<td>cryptographic subsystem</td>
</tr>
<tr>
<td>ASM</td>
<td>auxiliary storage manager (paging supervisor)</td>
</tr>
<tr>
<td>DYNPATH</td>
<td>path reconfiguration</td>
</tr>
<tr>
<td>SVC-33</td>
<td>I/O HALT I/O SVC routine</td>
</tr>
<tr>
<td>R-CLEAR</td>
<td>clear device recovery</td>
</tr>
<tr>
<td>R-SUBCHN</td>
<td>subchannel recovery</td>
</tr>
<tr>
<td>SVCPURGE</td>
<td>I/O PURGE I/O SVC routine</td>
</tr>
<tr>
<td>ALTPATH</td>
<td>alternate path recovery</td>
</tr>
<tr>
<td>MIH</td>
<td>missing interrupt handler</td>
</tr>
<tr>
<td>*UNKNOWN</td>
<td>unknown or unassigned driver code in use</td>
</tr>
</tbody>
</table>

Example 2

To display all outstanding I/O for device 283, type

TPIO 283

```
AMTI321 [4]  UNIT 283 PACK08 IOQ 003F8B IOSB 003F00 DVR EXCP
AMTI331 [5]  QDEPCH 5 RSV CNT 0 USE CNT 3 DCB CNT 2
```

Legend:
1. JES job ID of the address space using the device
2. name and current step for the address space using the device
3. dispatching priority for the address space
4. performance group and period
5. unit, volume serial number, IOQ and IOSB addresses of the I/O, and the I/O driver for this operation
6. number of I/Os waiting for access to this device
7. number of address spaces that have requested a reserve on this device
8. number of address spaces that have allocated this device
9. number of open DCBs against this device

**TQE**

The TQE service displays information about Disabled Interrupt Exit (DIE) routines scheduled for execution and pending timer interrupts for an address space.

**Syntax**

![Diagram](image)

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DIE       | indicates Disabled Interrupt Exit  
This operand displays the TQE chain for all DIEs (default).  
*Note:* A DIE executes as an extension to the first-level interrupt handler. Therefore, a DIE routine interrupts and executes under any dispatchable unit of work (TCB or SRB) for any address space. |
| asi       | address space instance; see “Identifying address spaces” on page 47 for more information |
Examples

To display the TQE chain for all DIEs, type

TQE

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Expire Time</td>
<td>ASID</td>
<td>Jobname</td>
<td>Address</td>
<td>Type</td>
<td>Flags</td>
<td>Creator</td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>AMTQE1I</td>
<td>14:04:37.67</td>
<td>0000</td>
<td>Unknown</td>
<td>01E91160</td>
<td>Real</td>
<td>SRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQE2I</td>
<td>14:04:37.69</td>
<td>0060</td>
<td>XTSTRPAS</td>
<td>90C09448</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE3I</td>
<td>14:04:37.76</td>
<td>0056</td>
<td>XTSTQPAS</td>
<td>925A8B20</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.77</td>
<td>0026</td>
<td>DC$PAS</td>
<td>927684D0</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE5I</td>
<td>14:04:37.78</td>
<td>0060</td>
<td>XTSTRPAS</td>
<td>925A8B20</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE6I</td>
<td>14:04:38.01</td>
<td>0001</td>
<td><em>MASTER</em></td>
<td>814F61A0</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE7I</td>
<td>14:04:38.19</td>
<td>0026</td>
<td>DC$PAS</td>
<td>9265B448</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE8I</td>
<td>14:04:38.32</td>
<td>0089</td>
<td>ORLMPROC</td>
<td>8F2F5A50</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE9I</td>
<td>14:04:38.58</td>
<td>0089</td>
<td>MTA8BILM</td>
<td>8EDBB710</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE10I</td>
<td>14:04:38.60</td>
<td>0056</td>
<td>XTSTQPAS</td>
<td>90DC74E8</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE11I</td>
<td>14:04:38.75</td>
<td>007E</td>
<td>MV50PMVS</td>
<td>8FD28448</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
<tr>
<td>AMTQE12I</td>
<td>14:04:38.99</td>
<td>008F</td>
<td>DEDDIRLM</td>
<td>8E071A50</td>
<td>Real</td>
<td>User</td>
<td>SetDIE</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1 expiration time for the TQE
2 hexadecimal address space ID
3 job name for the TQE
4 the address of the exit routine for the timer, which is the start of the code that is executed when the expire time is reached
5 type of TQE; can be REAL, TASK, WAIT, or UNKN
6 flags for the TQE, which can be one of the following types:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL</td>
<td>real TQE being times</td>
</tr>
<tr>
<td>USER</td>
<td>non-system TQE</td>
</tr>
<tr>
<td>DUMY</td>
<td>dummy system TQE</td>
</tr>
<tr>
<td>TLIM</td>
<td>time limit checking system TQE</td>
</tr>
<tr>
<td>SRM</td>
<td>system resources manager TQE</td>
</tr>
<tr>
<td>RME</td>
<td>RMF system TQE</td>
</tr>
<tr>
<td>MIDN</td>
<td>midnight system TQE</td>
</tr>
<tr>
<td>Flag</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>UNKN</td>
<td>unknown TQE</td>
</tr>
</tbody>
</table>

7 macro that created the TQE; it can be SETDIE, STIMER, STIMERM, or blank

8 TCB address for the TQE; a value is displayed only if the **TYPE** field contains TASK or WAIT

To display TQEs for address space JES2, type

```
TQE JES2
```

<table>
<thead>
<tr>
<th>AMTQE1I</th>
<th>Timer Queue Elements, Active at 14:08:22.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQE2I</td>
<td>Exit</td>
</tr>
<tr>
<td>AMTQE3I</td>
<td>Expire Time ASID Jobname Address Type Flags Creator Address</td>
</tr>
<tr>
<td>AMTQE8I</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:08:22.29 0019 JES2 8000B57E Real User Stimer 007FD098</td>
</tr>
</tbody>
</table>

**TRACK**

The TRACK (TRA) service provides a record of services invoked during the session and writes service output to a data set or SYSOUT.

**Syntax**

```
TRACK
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **xx**    | is a user-specified suffix  
You can track up to 32 services by placing the service names in a member in the BBPARM library data set. The member is named $INTKxx, where **xx** is the user-specified suffix. |
| **class** | specifies the SYSOUT class to which tracking output is routed |
| **dsname** | allows specification of a data set name to which output is routed  
The data set must be preallocated with a record format of VBA and a logical record length of 137. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>initiates tracking for all user-invoked services</td>
</tr>
<tr>
<td>START</td>
<td>initiates tracking for all user-invoked services</td>
</tr>
<tr>
<td>STOP</td>
<td>terminates tracking</td>
</tr>
<tr>
<td>FREE</td>
<td>deallocates tracking output</td>
</tr>
</tbody>
</table>

If routed to SYSOUT, tracking output is passed to JES for immediate printing.

**Examples**

- To initiate the recording of all services and their output to the preallocated data set *myid.output*, type
  
  \texttt{TRACK,START,myid.OUTPUT}

  \textit{Note}  
  
  A TRACK command can be included in the $$INTK xx member to specify the name of the output data set. If $$INTK xx contains a TRACK command, the first parameter is ignored if it is anything other than ALL. (Nesting is not permitted.)

- To initiate tracking of services specified in member $$INTK01$, type
  
  \texttt{TRACK,01,A}

  The output will be written to SYSOUT class A.

- To initiate tracking of all services and record the output in data set *myid.output*, type
  
  \texttt{TRACK,ALL,myid.OUTPUT}

- To stop tracking and free an output data set, type
  
  \texttt{TRACK,FREE}

- To stop tracking (leaving an output data set allocated), type
  
  \texttt{TRACK,STOP}

**$$INTKxx control statements**

A $$INTKxx member is read from the BBPARM data set under two conditions:

1. During initialization, when $$INTK00 is read and processed if present.
2. Whenever a two-character suffix (xx) is specified as the first parameter on a TRACK command.
A $INTKxx member can contain a TRACK command or a list of up to 32 services that are to be tracked. Each service must be listed on a separate line beginning in column 1.

When the $INTKxx member contains a TRACK command, the second parameter specifies the SYSOUT class or data set name for the tracking output. If the first parameter is a suffix, it is ignored. If the first parameter is ALL, all services are tracked.

**Note**

You cannot use a $INTKxx PARMLIB member to specify a list of services to be tracked when you are using the TRACK service from a MainView User Address Space (UAS). The TRACK service records the input and output for all services that you execute while tracking is active.

### Syntax rules

1. List each service on a separate statement.
2. Begin each entry in column 1.
3. Enter up to 32 services to be tracked.

### Examples

The following statement in $INTK00 will cause all services to be tracked and the tracking output to be written to a SYSOUT class T data set.

```
TRACK,ALL,T
```

The following statements will cause only the CSA and RSM services to be tracked. The tracking output will be written to data set MY.TRACK.DS. The TRACK command can be on any line. If multiple TRACK commands are present, only the last one will be used.

```
CSA
RSM
TRA,xx,MY.TRACK.DS
```

### Usage notes

- Tracking provides a record of command input and output. You can request tracking for all services or up to 32 specific services. You can direct the tracking output to a SYSOUT data set or an existing cataloged data set.

- The tracking facility is controlled by the TRACK service. See “TRACK” on page 264 for a description of this service.
Tracking output is formatted with headings and time stamps. The output format is VBA, and the logical record length is 137 bytes.

Tracking output is controlled by the second parameter on the TRACK command or by the TRACK DD statement if present. By default, output is written to SYSOUT class A.

**TSULIST**

The TSULIST (TSU) service displays TSO session information.

For active TSO users, the TSULIST service displays the following information:

- user ID
- address space ID
- line number
- type of system used
- terminal name

**Syntax**

```
TSULIST userid | userid *
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid</td>
<td>TSO user ID</td>
</tr>
<tr>
<td>userid*</td>
<td>partial TSO user ID</td>
</tr>
</tbody>
</table>

**Examples**

To display information about all TSO users currently logged on to the system, type

```
TSULIST
```

```
AMTTS1I USERID ASID NODE
```

Legend:

1  TSO user ID
2  ASID (in decimal format)
3  terminal name

To display information about TSO user BMCJMB2, type

**TSU,BMCJMB2**

<table>
<thead>
<tr>
<th>USERID</th>
<th>ASID</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCJMB2</td>
<td>240</td>
<td>TCPBL298</td>
</tr>
</tbody>
</table>

To display information about TSO users whose user IDs begin with the letters BMC, type

**TSU,BMC***

<table>
<thead>
<tr>
<th>USERID</th>
<th>ASID</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCJMB1</td>
<td>238</td>
<td>TCPBL299</td>
</tr>
<tr>
<td>BMCJMB2</td>
<td>240</td>
<td>TCPBL298</td>
</tr>
<tr>
<td>BMCGPR1</td>
<td>262</td>
<td>TCPBL119</td>
</tr>
<tr>
<td>BMCALV2</td>
<td>269</td>
<td>TCPD0041</td>
</tr>
<tr>
<td>BMCALV1</td>
<td>270</td>
<td>TCPD0044</td>
</tr>
</tbody>
</table>

To display information about TSO users whose user IDs begin with the letters BMC, type

**UCB**

The UCB service displays a unit control block (UCB) in hexadecimal and character formats.

In addition, important information contained in the UCB, such as device type, volume serial status, and path definition, is interpreted and presented in messages AMTU2AI, AMTU2BI, and AMTU2CI. You can specify either a device number or a volume serial number.

**Syntax**

```
UCB dvn volser
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dvn</code></td>
<td>is the device number whose UCB you want to display</td>
</tr>
<tr>
<td><code>volser</code></td>
<td>is the volume serial number you want to display</td>
</tr>
</tbody>
</table>

**Example**

To display the unit control block for volume serial number BAB901, type

```
AMTU2AI DVN=833F, Device class=DASD, Type=3390, Online, allocated, 64-Bit [1]
AMTU2BI Volser=BAB901, Status=Private, Shared [1]
AMTU2CI Defined paths=4, Active paths=4, LCU No=152 [1]
```

```
AMTU21I UCB 833F Relocated Prefix (First field is UCBRSTEM) [2]
AMTU21DI [7] [8] [9] [10]
AMTU25I 021D39A0 +0464 +X01D0 00040040 00000000 *........*
AMTU25I 021D39A8 +0472 +X01D8 00000000 000128E7 *........*
AMTU25I 021D39B0 +0480 +X01E0 289C0CCF B00080F0 *........0*
AMTU25I 021D39B8 +0488 +X01E8 70784432 00000000 *........*
AMTU25I 021D39C0 +0496 +X01F0 01080000 00000001 *........*
```

```
AMTU21II UCB 833F Prefix [3]
AMTU25I 00EFBD38 -0008 -X0008 00000000 01A90FB0 *......z..*
```

```
AMTU22I UCB 833F Common Section [4]
AMTU25I 00EFBD40 +0000 +X0000 0088FFBC 833F0000 *...h..c...*
AMTU25I 00EFBD48 +0008 +X0008 00000000 00E4E3C2 *.....UCB*
AMTU25I 00EFBD50 +0016 +X0101 30F0200F 00EFBD18 *........*
```

```
AMTU23I UCB 833F Device Dependent Section [5]
AMTU25I 00EFBD58 +0024 +X0101 002D0100 C21C2F9 *.B..BA9*
AMTU25I 00EFBD60 +0032 +X0020 F0F11000 00000003 *01......*
AMTU25I 00EFBD68 +0040 +X0028 38680076 13142380 *........*
```

```
AMTU24I UCB 833F Common Extension [6]
AMTU25I 00EFBD18 +0000 +X0000 00009090 20AA0008 *........*
AMTU25I 00EFBD20 +0008 +X0008 021D39A0 000000DC *........*
AMTU25I 00EFBD28 +0016 +X0010 00000000 00F0DE94 *........m*
AMTU25I 00EFBD30 +0024 +X0018 00EFBCE0 00000000 *........*
```

Legend:

1 interpreted information from UCB

2 UCB relocated prefix section; the relocated prefix is a data area in the portion of the UCB prefix relocated above the 16-MB line

3 UCB prefix section for the device

4 UCB common section for the device

5 UCB device-dependent section for the device

The length of this section is dependent upon the type of device represented by the UCB.
6 UCB common extension for this device
The length of this section is dependent upon the type of device represented by the UCB.

7 address in hexadecimal format

8 offset from the origin in decimal and hexadecimal format

9 hexadecimal representation of data (eight bytes per line)

10 character representation of data (eight bytes per line)

**USING**

The USING (USI) service displays the active address spaces that are using a specified device.

For example, if a tape drive appears inactive, you can determine which address space has allocated the device.

**Syntax**

```
USING dvn volser [IN | ALL]
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dvn</code></td>
<td>is the hexadecimal device number</td>
</tr>
<tr>
<td><code>volser</code></td>
<td>is a complete six-character volume serial number</td>
</tr>
<tr>
<td>IN</td>
<td>displays only the swapped-in address spaces that are using the device (default)</td>
</tr>
<tr>
<td>ALL</td>
<td>displays all the active address spaces that are using the device</td>
</tr>
</tbody>
</table>

**Example**

To display the active address spaces that are using device BAB329, type
**USING BAB329**

```
13:01:47 CMD=USING_BAB329
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTD29</td>
<td>Unit Job Name DDNAME Data Set Name</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MQMPASRK STEPLIB MQS12.BASE.LINKLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MQMPASRK BBLOAD MQS12.BASE.LINKLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MQMPASRK ISPPROF BITRKK.XDC2.ISPPROF</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F DDH1X BBIPROF DDH1.BBPROF</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F DDH1X IPCSDDIR IPCS.DDH1ESA.IPCSDDIR</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F X18HGBG DFSOLP03 IMS.V5100.OLP03</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F X18HGBG MATRIXB IMS.V5100.MATRIXB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MMQJB120 STEPLIB MQS12.BASE.LINKLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MMQJB120 BBLOAD MQS12.BASE.LINKLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F MMQJB120 DBGLIB BBM33.ENG.DBGLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F BMVSLK2 BBCLIB BB.XTSTH.BBCLIB</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F BMVSLK2 BBIPROF BMVSLK.BMVSLK2.BBPROF</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F AAOSSDHC BBIPARM DDH1.DH31.TRAVEL</td>
<td></td>
</tr>
<tr>
<td>AMTD27</td>
<td>D3F BOLGBG3 STEPLIB IPX11.BOLGBG.LOAD</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**

1. job name
2. DD name of the DD statement for the data set allocated to this device
3. data set name

---

**VIO**

The VIO service displays usage of VIO slots by job name or user ID.

### Syntax

![VIO Diagram]

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>threshold</code></td>
<td>is a one- to four-digit integer; only address spaces that use a number of VIO slots greater than this value are listed; the default is 0</td>
</tr>
</tbody>
</table>

### Examples

To display the number of VIO slots used by each job name or user ID, type
VIO

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVI1 VMON1  HAS  2 VIO SLOT(S) IN USE</td>
<td></td>
</tr>
<tr>
<td>AMTVI1 SUP6   HAS  5 VIO SLOT(S) IN USE</td>
<td></td>
</tr>
<tr>
<td>AMTVI1 TESTJB2 HAS  12 VIO SLOT(S) IN USE</td>
<td></td>
</tr>
<tr>
<td>AMTVI1 VMON8  HAS  8 VIO SLOT(S) IN USE</td>
<td></td>
</tr>
<tr>
<td>AMTVI1 VMON9  HAS  2 VIO SLOT(S) IN USE</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1 job name or user ID

2 number of VIO slots used

To display user IDs or job names that are using 45 or more VIO slots, type VIO 45

AMTVI3I NO JOBS USING 45 OR MORE VIO SLOT(s)

VMCMD

The VMCMD (VMC) service lets an z/OS user running as a VM guest issue CP commands through the virtual console interface.

Syntax

```
VMCMD any valid CP command
cpcmd
```
A pound sign (#) delimits multiple CP commands.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpcmd</td>
<td>any valid CP command</td>
</tr>
</tbody>
</table>

Example

To enter the CP command Q NAMES, type

```
VMCMD Q NAMES
```

| AMTVI1 TOMCFC   - DSC , MVSSJSI - DSC , COP5 - 1321 , 4ESAMAP - DSC |
| AMTVI1 FESAMAP  - DSC , LESAMAP - DSC , 5ESAMAP - DSC , RESAMAP - DSC |
| AMTVI1 CESAMAP  - DSC , MVSAPDI - DSC , 50PER2 - 1322 , HTTPD - DSC |
| AMTVI1 FINGERD  - DSC , 5UFTD - DSC , IDENTD - DSC , 5FTPSRV - DSC |
| AMTVI1 BPUTIL   - DSC , TCP5VS1 - DSC , TCPIP - DSC , TROTH - DSC |
| AMTVI1 5RSCS    - DSC , 5VTAM - DSC , VM5TCPPIP - DSC , 5BATCH - DSC |
| AMTVI1 5RTM     - DSC , XDIR - DSC , 5GCS - DSC , 5ESAWRIT - DSC |
| AMTVI1 5VTMPROP - DSC , WRKRBEE - DSC , 5TCPPROP - DSC , 5ESASERV - DSC |
| AMTVI1 APDFCC   - DSC , SDACFCG - DSC , CFGONSVR - DSC , CHKSTORE - DSC |

272  MainView SYSPROG Services User Guide and Reference
The WARNING (WAR) service (alias AEW) provides the ability to start and stop the Exception Monitor and display and alter the parameters for active samplers.

For an explanation of the various samplers and their parameters, see “Exception Monitor samplers” on page 281.

**Note**

The WARNING (AEW) service is not supported when MainView SYSPROG Services is executed as a TSO command.

### Syntax

![Syntax Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>lists the Exception Monitor samplers currently active (default)</td>
</tr>
</tbody>
</table>
| START     | starts the Exception Monitor system  
If `xx` is not specified, BBPARM member PWSCPM00 is used. |
| `xx`      | is the suffix used to form the name of the BBPARM member (PWSCPM `xx`) that contains the sampler control statements to be used to start the Exception Monitor  
The default member name is PWSCPM00. |
| STOP      | stops the Exception Monitor |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATS</td>
<td>displays a status report detailing the CPU time used by the Exception Monitor. This report is also displayed when the Exception Manager is stopped.</td>
</tr>
<tr>
<td>samp(n)</td>
<td>WARNING samp causes all parameters and their associated values for sampler samp to be displayed. (n) is optional and refers to the nth sampler with the name samp. The default value is 1. See examples 6 and 7.</td>
</tr>
<tr>
<td>parm</td>
<td>WARNING samp, parm causes the value associated with parameter parm for sampler samp to be displayed.</td>
</tr>
<tr>
<td>PARM=value</td>
<td>WARNING samp, parm=value causes parameter parm for sampler samp to be set to value value. You can specify multiple parameters and values for the sampler on a single WARN command. You can also use scaled numeric values. See example 7.</td>
</tr>
</tbody>
</table>

### Example 1

To display the active Exception Monitor samplers, type

**WARN**

```
WARN
AMTH21 The following AEW Samplers are active:
AMTTH3I INT     DEV     DNR     MTP
```

### Example 2

To start the Exception Monitor, type

**WARN,START,01**

```
WARN,START,01
PWSSDV02 AEW Initializing
PWSSIN010 AEW parameters obtained from member PWSCPM01
PWSSDV65 AEW Monitor Active, 5:00:24 PM, SJSD, Release: 3.6.00
PWSSDV09 AEW Initialization Complete
AMTAW11 AEW initialization completed
```

### Example 3

To display the status report, type

**WARN,STATS**

```
AMTV31I                        TCB Time  Rate
AMTV32I  Sampler  Calls  Microsec  Microsec  Percent
AMTV33I  ---------  --------  --------  --------  --------
AMTV34I  GBLS      184     5,231        28       0.24
AMTV34I  JSU       184     2,039K   11,083      95.06
AMTV34I  Admin     -N/A-      101K      273       4.68
AMTV33I  ---------  --------  --------  --------  --------
AMTV34I  Total/Avg 368     2,145K    5,829     100.00
AMTV35I Average absorption rate 1.400 seconds per hour
AMTV36I  AEW DIE Monitor Statistics
AMTV37I                     Total   Nano Secs
AMTV38I  DIE    Samples   Mic Sec  Per sample
AMTV38I                      -------  --------  --------
```

---

274 MainView SYSPROG Services User Guide and Reference
Example 4

To display all of the parameters and their associated values for the DNR sampler, type

`WARN,DNR`

`WARN,DNR`

`AMTTL8I DNR parameter value(s):
AMTW51I INT = 1
AMTW51I NOTREADY = 2`

Example 5

To display the value associated with PENDING parameter for the MTP sampler, type

`WARN,MTP,PENDING`

`WARN,MTP,PENDING`

`AMTW51I PENDING = 3`

Example 6

To change the value of the PENDING parameter for the MTP sampler to 5, type

`WARN,MTP,PENDING=5`

`WARN,MTP,PENDING=5`

`AMTW51I PENDING = 5`

Example 7

This example shows how to use the WARNING service to change the minimum service units rate threshold (`minrate`) for the third BSRV sampler to 10,000 service units per second.

Because there are multiple BSRV samplers and the parameters for the third BSRV sampler needs updating, you must add the suffix (3) to the sampler name (BSRV(3)). The rate is a numeric value and you can specify a scaled value. In this example, the scaled value 10K represents 10,000 service units per second. This example also shows that the report interval value is 2 (INT=2).

You can specify both the rate and interval parameters at the same time; enter the following command to receive the following results.

`WARN,BSRV(3),RATE=10K,INT=2`

`AMT000I WARN,BSRV(3),MINRATE=10K,INT=2`

`AMTW51I MINRATE = 10,000`

`AMTW51I INT = 2`
WHERE

The WHERE (WHE) service displays information about the storage area containing the subject address and where it is located within that storage area.

The WHERE service has functionally replaced the LPA and NUCLEUS services.

Syntax

```
WHERE - expression  asi
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>is any expression that resolves to a storage address, any module name within the Link Pack Area (LPA), or any control section name within the nucleus. See “Expressions” on page 48 for more information.</td>
</tr>
<tr>
<td>asi</td>
<td>address space instance; see “Identifying address spaces” on page 47 for more information</td>
</tr>
</tbody>
</table>

**Note**
The second parameter is required only if the expression is not associated with an address space or if you want to associate the expression with a different address space.

**Example 1**
To locate the JSCB control block for the master address space and display the storage subpool and key, type

```
WHERE JSCB,(1)
```

```
AMTWH4I  007F2F54  Found in Local System Queue Area (LSQA) or SWA
AMTWH7I          Subpool 230, key 0p
```

**Note**
The lowercase p after the subpool number indicates that the storage is fetch protected.

**Example 2**
To locate address 0889F010, type

```
WHERE - expression  asi
```
Example 3

To determine the location of the storage referenced by the double-word address constant at offset 60 in memory object MOBJ0001, type

```
WH,MOBJ0001+60!
```

The exclamation mark (!) is the indirection symbol for 64-bit addresses. The expression indicates that information is desired about the storage referenced by the 64-bit address at +60 within memory object MOBJ0001.

**Note**

Symbol MOBJ0001 must have been previously defined. Use the OLIST service to locate and create symbols for all memory objects in an address space.

Example 4

To determine the location of module IEFBR14, type

```
WH,IEFBR14
```

Usage notes

The first parameter is required and can be

- any expression
- name of a module within the LPA
- name of a control section within the system nucleus

The second parameter is optional. It is required only if the expression is not associated with an address space or you want it to be associated with a specific address space. The WHERE command supports all virtual memory (both below and above the bar).

ZAP

The ZAP service modifies main storage in any address space.
Memory objects are also supported.

**Syntax**

```
ZAP  expression  vvvvvvvv  rrrrrrrr  asi
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>see “Expressions” on page 48</td>
</tr>
</tbody>
</table>
| vvvvvvvv   | is the Verify data field  
You can change up to 16 bytes of data at a time. Specify the data in an even number of hexadecimal digits, from 2 to 32. You must verify all replaced data. |
| rrrrrrrr   | is the Replacement data field  
You can change up to 16 bytes of data at a time. Specify the data in an even number of hexadecimal digits, from 2 to 32. The Replacement field must equal the Verify field in length. You must verify all replaced data. |
| asi        | address space instance; see “Identifying address spaces” on page 47 for more information |

**Example**

To modify location 02F1C in the LLA address space from D2033008CA22 to 070007000700, type

```
ZAP 02F1C D2033008CA22 070007000700 LLA
```

**Usage notes**

- You can omit the address space instance parameter if the hexadecimal location is in common storage (CSA, LPA, SQA, or the nucleus) or is implied by the expression. See “Expressions” on page 48 for more information.

- Changes made by using the ZAP service are effective only for the duration of the IPL. The ZAP service page-fixes changes to locations in the following areas:
  - EPLPA (Extended Pageable Link Pack Area)
  - EMLPA (Extended Modifiable Link Pack Area)
  - MLPA (Modifiable Link Pack Area)
— PLPA (Pageable Link Pack Area)

Changes to locations are page-fixed by ZAP to prevent the changes from being overwritten when page-ins occur.

- For processor complexes consisting of more than one CPU, you can specify a CPU number. Add `/c` to the end of the value specified for *hex location*, where `c` is the single hexadecimal number (0 through 9) of a specific CPU. Use this field when you ZAP the Prefixed Save Area (PSA).

### ZIIP

The ZIIP (ZII) service displays information about actual and potential IBM System z Integrated Information Processor (zIIP) usage for each address space that has work eligible for execution on zIIPs.

#### Syntax

There are no parameters for the ZIIP service.

#### Example

Type

ZIIP

<table>
<thead>
<tr>
<th>AMTII4I</th>
<th>Adrs space</th>
<th>CPU Milli-sec</th>
<th>zIIP Actual</th>
<th>zIIP Eligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTII5I</td>
<td>------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>AMTII6I</td>
<td>CMFJAVA</td>
<td>2,394,969</td>
<td>47.69%</td>
<td>0.12%</td>
</tr>
<tr>
<td>AMTII6I</td>
<td>CMFJAVA2</td>
<td>219,121</td>
<td>8.70%</td>
<td>0.07%</td>
</tr>
<tr>
<td>AMTII3I</td>
<td>Located 2 address spaces with zIIP time</td>
<td>[5]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

1. name of address space
2. total CPU usage
3. actual zIIP processor percentage of the total processor (CPU) usage
4. potential (eligible) zIIP processor percentage of the total processor (CPU) usage
5. number of AMTII6I messages (one per address space)
Exception Monitor samplers

This section provides detailed information about the Exception Monitor samplers, including proper syntax, required and optional parameters, and usage examples.

## Introduction

The Exception Monitor helps you detect potential problems by monitoring thresholds that you establish and sending a warning message when a threshold is exceeded.

You can use the Exception Monitor in the following ways:

- **When MainView SYSPROG Services is operating as a batch job or started task**
  When the Exception Monitor is operating as a batch job or a started task, warning messages are sent to the console.

- **From the MainView for z/OS product**
  When you use the Exception Monitor from the MainView for z/OS product, you can access warning messages from the WARN view and you can send warning messages (optionally) to the console.
  
  For more information, see the description for the LCLAEW parameter in “$INSYS0” on page 24.

- **From the MainView AutoOPERATOR for z/OS product option**

In all environments, you can also specify that warning messages are directed to the AOAnywhere API in MainView AutoOPERATOR. For more information, see the description for the AEWALERT parameter in “$INSYS0” on page 24.

There are two ways to control the Exception Monitor samplers:

- **Using the WARNING Service**, which is described in this publication
Through the EMSTAT View, which is described in *MainView for z/OS User Guide and Reference*

In all environments, you must add to PARMLIB member PWSCPMxx the Exception Monitor samplers and their parameters that you want to use, where xx is a user-defined two-character suffix. For more information, see the description for the AEWSTART parameter in “$$INSYS0” on page 24.

You can use the WARNING service to dynamically change individual parameters on active samplers. For more information, see “WARNING” on page 273.

**ACPU**

The Address Space CPU (ACPU) sampler produces a warning message when an address space uses more than a specified percentage of CPU time during the reporting interval and the total CPU utilization has exceeded the specified percentage.

*Note*

You can suppress reporting for specific address spaces by including their names in BBPARAM member $$INEXCL. For more information, see “$$INEXCL” on page 23.

**Syntax**

```
ACPU 1 name percent
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| int       | is the length of the sample period in intervals  
The default sample period is 30 seconds (one interval). |
| name      | is the name of the address space to be monitored  
If the last character of the name is an asterisk (*), the name is to be treated as a generic address-space name prefix; that is, any address space whose name matches the character string up to but not including the asterisk is monitored.  
To monitor all address spaces, except those in the exclusion list, specify an asterisk (*). |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{percent}</td>
<td>is the CPU busy percentage for which an address space must be responsible before an exception message is issued for that address space</td>
</tr>
<tr>
<td>\textit{type}</td>
<td>is the type of address space to be monitored, where</td>
</tr>
<tr>
<td>&amp; B indicates a batch job</td>
<td></td>
</tr>
<tr>
<td>&amp; S indicates a Started Task</td>
<td></td>
</tr>
<tr>
<td>&amp; T indicates a TSO user</td>
<td></td>
</tr>
<tr>
<td>&amp; A indicates any address space matching the specified name; the default</td>
<td></td>
</tr>
<tr>
<td>\textit{minpct}</td>
<td>is the percentage of system CPU busy that must be reached before exception messages are issued</td>
</tr>
<tr>
<td>&amp; The default is a system CPU busy percent greater than 0.</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example uses a reporting interval of 30 seconds. The example control statement issues the PWSACP94 warning message (at two-minute intervals) for each address space that is not in the exception list that had at least a 10% CPU utilization and if the total CPU utilization during the interval was at least 50%.

\texttt{ACPU 4,*,10,A,50}

### Warning message

The following warning message is issued by the ACPU sampler:

\texttt{PWSACP94 *WARNING* CPU usage is percent \% for type name}

The variables in the message are:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{percent}</td>
<td>is the percentage of cpu time that the address space used</td>
</tr>
<tr>
<td>\textit{type}</td>
<td>is the type of address space (Job, STC, TSU, or All)</td>
</tr>
<tr>
<td>\textit{name}</td>
<td>is the name of the address space exceeding the limit</td>
</tr>
</tbody>
</table>

### AIO

The Address Space I/O (AIO) sampler produces a warning message when an address space uses more than a specified percentage of the total system I/O activity during a reporting interval.
You can suppress the message if a given total system I/O rate threshold is not reached during a reporting interval.

**Note**
You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals&lt;br&gt;The default sample period is 30 seconds (one interval).</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the address space to be monitored&lt;br&gt;If the last character of the name is an asterisk (*), the name is to be treated as a generic address-space name prefix; that is, any address space whose name matches the character string up to but not including the asterisk is monitored.</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system I/O for which an address space must be responsible before an exception message is issued for that address space&lt;br&gt;If this value is zero, no warning messages will be issued.</td>
</tr>
</tbody>
</table>
| type      | is the type of address space to be monitored, where<br>  
  B indicates a batch job<br>  S indicates a Started Task<br>  T indicates a TSO user<br>  A indicates any address space matching the specified name; the default |
| rate      | is the rate of EXCPs per second that must be reached before exception messages are issued<br>The default is a system I/O rate greater than 25 EXCPs per second. |

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the system I/O rate exceeds 1,000 EXCPs-per-second over the preceding two minutes:
AIO 4,*,20,A,1000
Also, a warning is issued for any address space doing more than 20 percent of the I/O.

Warning messages
The following warning messages are issued by the AIO sampler:

PWSAI090 *WARNING* TOTAL SYSTEM I/O RATE IS ppppp EXCPs/SEC

Message PWSAI090 indicates that the system-I/O rate has exceeded the warning threshold.

PWSAI092 *WARNING* I/O RATE FOR ttt nnnnnnnn IS xxx% OF TOTAL

Message PWSAI092 indicates that the specified address space has exceeded the warning threshold for percentage of system I/O, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is STC, TSU, or JOB</td>
</tr>
<tr>
<td>nnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percent of the total system I/O rate</td>
</tr>
</tbody>
</table>

AIOR
The Address Space I/O Rate (AIOR) sampler produces a warning message when an address space issues EXCPs at a rate that exceeds the defined threshold during a reporting interval.

Note
You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 23.

Syntax

\[
\text{AIOR} \quad \text{int} \quad \text{name} \quad \text{rate} \quad \text{type} \quad >25 \quad A
\]

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals</td>
</tr>
</tbody>
</table>
| name      | is the name of the address space to be monitored  
If the last character of the name is an asterisk (*), the name is to be treated as a generic address-space name prefix; that is, any address space whose name matches the character string up to but not including the asterisk is monitored. |
| rate      | is the rate of EXCPs-per-second that must be reached before exception messages are issued  
The default is an I/O rate greater than 25 EXCPs per second. |
| type      | is the type of address space to be monitored, where  
B indicates a batch job  
S indicates a Started Task  
T indicates a TSO user  
A indicates any address space matching the specified name; the default |

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals for each address space whose I/O rate exceeds 25 EXCPs-per-second over the preceding two minutes:

\[ \text{AIOR 4,*,25,A} \]

**Warning message**

The following warning message is issued by the AIOR sampler:

\[ \text{PWSAIR91 *WARNING* I/O RATE FOR ttt nnnnnnnn IS excep EXCPs/SEC} \]

Message PWSAIR01 indicates that the specified address space has exceeded the warning threshold for I/O rate, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is STC, TSU, or JOB</td>
</tr>
<tr>
<td>nnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>excep</td>
<td>is the EXCPs issued by the address space during the interval</td>
</tr>
</tbody>
</table>
**APAG**

The Address Space Paging (APAG) sampler produces a warning message when an address space uses more than a specified percentage of the total system paging activity during a reporting interval.

You can suppress the message if a given total system paging-rate threshold is not reached during the reporting interval.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals. The default sample period is 30 seconds (one interval).</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the address space to be monitored. If the last character of the name is an asterisk (*), the name is to be treated as a generic address-space name prefix; that is, any address space whose name matches the character string up to but not including the asterisk is monitored.</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system paging for which an address space must be responsible before an exception message is issued for that address space.</td>
</tr>
<tr>
<td>type</td>
<td>is the type of address space to be monitored, where B indicates a batch job S indicates a Started Task T indicates a TSO user A indicates any address space matching the specified name; the default</td>
</tr>
<tr>
<td>syspag</td>
<td>is the paging rate-per-second that must be reached before exception messages are issued. The default is a system paging rate greater than 25 pages per second.</td>
</tr>
</tbody>
</table>
Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the system-paging rate exceeds 50 pages-per-second over the preceding two minutes:

```
APAG 4,*,35,A,50
```

A warning is issued for any address space doing more than 35 percent of the paging.

Warning messages

The following warning messages are issued by the APAG sampler:

```
PWSAPAG0 *WARNING* TOTAL SYSTEM PAGING RATE IS xxx PAGES/SEC
```

Message PWSAPAG0 indicates that the system-paging rate has exceeded the warning threshold.

```
PWSAPAG1 *WARNING* PAGING RATE FOR ttt nnnnnnnn IS xxx% OF SYSTEM TOTAL
```

Message PWSAPAG1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is STC, TSU, or JOB</td>
</tr>
<tr>
<td>nnnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percentage of the total system paging rate</td>
</tr>
</tbody>
</table>

APGR

The Address Space Paging Rate (APGR) sampler produces a warning message when an address space pages at an excessive rate during a reporting interval.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| int       | is the length of the sample period in intervals  
The default sample period is 30 seconds (one interval). |
| name      | is the name of the address space to be monitored  
If the last character of the name is an asterisk (*), the name is to be treated as a generic address-space name prefix; that is, any address space whose name matches the character string up to but not including the asterisk is monitored. |
| pgrate    | is the paging rate-per-second that must be reached by an address space before exception messages are issued  
The default is an address-space paging rate greater than 10 pages per second. |
| type      | is the type of address space to be monitored, where  
B indicates a batch job  
S indicates a Started Task  
T indicates a TSO user  
A indicates any address space matching the specified name; the default |

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if any address-space paging rate exceeds 35 pages-per-second over the preceding two minutes:

APGR 4,*,35,a

Warning message

The following warning message is issued by the APGR sampler:

PWSAPGR1 *WARNING* PAGING RATE FOR ttt nnnnnnn IS ppp PAGES/SEC

Message PWSAPGR1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where
### ASRV

The Address Space Service Rate Percentage (ASRV) sampler produces a warning message when an address space uses more than the specified percentage of one or more SRM service component resources during a reporting interval.

The message can be suppressed if a given total system service rate threshold is not reached during the reporting interval.

You can use multiple ASRV sampler statements; for example, you might want to have one TSRV statement for each measure.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

### Syntax

![Syntax diagram](image)

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals</td>
</tr>
<tr>
<td></td>
<td>The default sample period is 30 seconds (one interval).</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the address space to be monitored</td>
</tr>
<tr>
<td></td>
<td>If the last character of the name is an asterisk (*), the name is to be</td>
</tr>
<tr>
<td></td>
<td>treated as a generic address-space name prefix; that is, any address space</td>
</tr>
<tr>
<td></td>
<td>whose name matches the character string up to but not including the asterisk</td>
</tr>
<tr>
<td></td>
<td>is monitored.</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of the SRM service component resources being monitored</td>
</tr>
<tr>
<td>type</td>
<td>indicates the type of service component being monitored</td>
</tr>
<tr>
<td>measure</td>
<td>specifies the measure to be monitored</td>
</tr>
<tr>
<td>minrate</td>
<td>specifies the minimum rate to be monitored</td>
</tr>
</tbody>
</table>

---

**Variable**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ttt</strong> is STC, TSU, or JOB</td>
</tr>
<tr>
<td><strong>nnnnnnnn</strong> is the address space name</td>
</tr>
<tr>
<td><strong>ppp</strong> is the number of pages per second</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>percent</strong></td>
<td>is the percentage of the total system service rate that an address space must be responsible for before an exception message is issued for that address space</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>is the type of address space to be monitored, where</td>
</tr>
<tr>
<td></td>
<td>B indicates a batch job</td>
</tr>
<tr>
<td></td>
<td>S indicates a Started Task</td>
</tr>
<tr>
<td></td>
<td>T indicates a TSO user</td>
</tr>
<tr>
<td></td>
<td>A indicates any address space matching the specified name; the default</td>
</tr>
<tr>
<td><strong>measure</strong></td>
<td>is the SRM component use rate to be monitored:</td>
</tr>
<tr>
<td></td>
<td>C indicates CPU service units</td>
</tr>
<tr>
<td></td>
<td>I indicates I/O service units</td>
</tr>
<tr>
<td></td>
<td>M indicates MSO service units</td>
</tr>
<tr>
<td></td>
<td>T indicates total service units; the default</td>
</tr>
<tr>
<td><strong>minrate</strong></td>
<td>is the system SRM-resource service rate (in service units per second) that must be reached before exception messages are issued</td>
</tr>
<tr>
<td></td>
<td>The default is a system service rate greater than zero SUs per second.</td>
</tr>
</tbody>
</table>

### Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals whenever an address space being monitored uses more than the 35 percent of total SRM service components during a given reporting interval:

\[ \text{ASRV 4,*,35,A,T,50} \]

You can suppress a message if a given total system service rate threshold of 50 is not reached during a reporting interval.

### Warning messages

The following warning message is issued by the ASRV sampler:

\[ \text{PWSASRV1 *WARNING* mmm SERVICE RATE FOR ttt jjjjjjjjj IS ppp\% OF nnn SUs} \]

Message PWSASRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmm</td>
<td>is the SRM service component being monitored</td>
</tr>
<tr>
<td>ttt</td>
<td>is STC, TSU, or JOB</td>
</tr>
</tbody>
</table>
### ASSR

The Address Space Service Unit Rate (ASSR) sampler produces a warning message when an address space uses SRM service components at a rate that exceeds the defined threshold during a reporting interval.

The warning will be suppressed if the specified minimum total system-service rate (minrate) is not reached during a reporting interval.

#### Note

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

### Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the address space to be monitored</td>
</tr>
<tr>
<td>rate</td>
<td>is the SRM component service rate for an address space that must be reached before exception messages are issued</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>is the type of address space to be monitored, where</td>
</tr>
<tr>
<td></td>
<td>B indicates a batch job</td>
</tr>
<tr>
<td></td>
<td>S indicates a Started Task</td>
</tr>
<tr>
<td></td>
<td>T indicates a TSO user</td>
</tr>
<tr>
<td></td>
<td>A indicates any address space matching the specified name; the default</td>
</tr>
<tr>
<td><strong>measure</strong></td>
<td>is the SRM service measure to be monitored:</td>
</tr>
<tr>
<td></td>
<td>C indicates CPU service</td>
</tr>
<tr>
<td></td>
<td>I indicates I/O service</td>
</tr>
<tr>
<td></td>
<td>M indicates MSO service</td>
</tr>
<tr>
<td></td>
<td>T indicates total service; the default</td>
</tr>
<tr>
<td><strong>minrate</strong></td>
<td>is the minimum total service rate that must be reached to allow warning messages to be issued</td>
</tr>
</tbody>
</table>

**Example**

Assuming an interval of 30 seconds, the following control statement displays warning messages at two-minute intervals whenever any address space uses more than a total of 100 SRM service units during a given reporting interval:

```
ASSR 4*,100,a,t
```

**Warning message**

The following warning message is issued by the ASSR sampler:

```
PWSASSR1 *WARNING* mmm SERVICE RATE FOR ttt IS rrrrr SU/SEC
```

Message PWSASSR1 indicates that the address-space service-unit rate has exceeded the warning threshold, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mmm</strong></td>
<td>is the SRM service component being monitored</td>
</tr>
<tr>
<td><strong>ttt</strong></td>
<td>is STC, TSU, or JOB</td>
</tr>
<tr>
<td><strong>jjjjjjj</strong></td>
<td>is the address space name</td>
</tr>
<tr>
<td><strong>rrrrr</strong></td>
<td>is the number of service units</td>
</tr>
</tbody>
</table>
The ASVT sampler produces a warning message if the percentage of slots on any of the three queues is less than the specified threshold value.

When programs that establish PCs terminate, the address space ID is not reusable until all address spaces that had established a cross-system bind with the address space terminate. When this situation occurs, a slot is taken off of the Non-reusable Replacement Queue to maintain the MAXUSER value.

After the Non-reusable Replacement Queue is depleted, the maximum number of address spaces (number on the available queue plus the number active) is decreased each time an address space becomes not-reusable. Eventually, the Available Queue might be depleted, preventing the start of additional address spaces. The only exception is that Started Tasks can be initiated as long as the Start/SASI Queue is not depleted.

Depletion of these queues might force a system IPL. Therefore, monitoring the queues might give you an opportunity to take corrective action to avoid an IPL.

### Syntax

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the report period in intervals; the default is 30 seconds (one interval)</td>
</tr>
<tr>
<td>avq</td>
<td>is the threshold percentage for the Available Queue</td>
</tr>
<tr>
<td></td>
<td>A warning message is displayed when the number of slots on the Available Queue, taken as a percentage of the original number, is less than the specified threshold value.</td>
</tr>
<tr>
<td>rplq</td>
<td>is the threshold percentage for the Non-reusable Replacement Queue</td>
</tr>
<tr>
<td></td>
<td>A warning message is displayed when the number of slots on the Non-reusable Replacement Queue, taken as a percentage of the original number, is less than the specified threshold value.</td>
</tr>
<tr>
<td>strtq</td>
<td>is the threshold percentage for the Start/SASI Queue</td>
</tr>
<tr>
<td></td>
<td>A warning message is displayed when the number of slots on the Start/SASI Queue, taken as a percentage of the original number, is less than the specified threshold value.</td>
</tr>
</tbody>
</table>
```
**Example**

ASVT,4,20,50,80
The reporting period is four intervals (two minutes if the default interval is used). A warning message will be issued if one of the following occurs:

- Available Queue is less than 20% of the USERMAX value.
- Non-reusable Replacement Queue is less than 50% of the RSVNONR value.
- Start/SASI Queue is less than 80% of the RSVSTRT value specified when an IPL was performed on the system.

**Warning message**

The following warning messages are issued by the ASVT sampler:

- **PWSASVT1** *WARNING* Available Queue is ***xxx*** % of MAXUSER
  
  Message PWSASVT1 is issued when result of the following equation exceeds the AVQ (Available Queue) percentage specified for the ASVT sampler in your BBPARM member PWSCPMxx:
  
  \[
  \left( \frac{\text{total-slots-available}}{\text{slots-in-use}} \right) \times 100
  \]

- **PWSASVT2** *WARNING* Non-Reusable Replacement Queue is ***xxx*** % of RSVNONR
  
  Message PWSASVT2 is issued when the result of the following equation exceeds the RPLQ (Non-reusable Replacement Queue) percentage specified for the ASVT sampler in your BBPARM member PWSCPMxx:
  
  \[
  \left( \frac{\text{reserved-slots-available}}{\text{remaining-reserved-slots}} \right) \times 100
  \]

- **PWSASVT3** *WARNING* START/SASI queue is ***xxx*** % of RSVSTRT
  
  Message PWSASVT3 is issued when the result of the following equation exceeds the STRTQ (STC Reserved Queue) percentage specified for the ASVT sampler in your BBPARM member PWSCPMxx:
  
  \[
  \left( \frac{\text{reserved-STC-slots}}{\text{remaining-STC-slots}} \right) \times 100
  \]
The message can be suppressed if a given threshold for total CPU usage is not reached during the reporting interval.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

![Diagram](image)

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system CPU time that a batch address space must be responsible for before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of address spaces to be listed whenever a CPU overload condition is detected; the default is 5</td>
</tr>
<tr>
<td>minpct</td>
<td>is the CPU busy percentage that must be reached before exception messages are issued; the default is a system CPU busy percentage greater than zero</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays up to five warning messages at two-minute intervals if the system-CPU busy percentage exceeds 50 percent over the preceding two minutes:

**BCPU 4,35,5,5**

A warning is issued for any batch address space that has been active 35 percent of the total CPU busy time.

**Warning message**

The following warning message is issued by the BCPU sampler:

```
PWSBCPU1 *WARNING* CPU USAGE IS ppp% FOR JOB jjjjjjjjjj
```
Message PWSBCPU1 indicates that the specified address space has exceeded the warning threshold for percentage of system-CPU busy time, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppp</td>
<td>is the address space busy percentage</td>
</tr>
<tr>
<td>jjjijiji</td>
<td>is the address space name</td>
</tr>
</tbody>
</table>

**BIO**

The Batch I/O (BIO) sampler produces a warning message when a batch address space uses more than a specified percentage of the total system I/O activity during a reporting interval.

The message can be suppressed if a given total system I/O rate threshold is not reached during a reporting interval.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $\$INEXCL$. For details, see “$\$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length in intervals of the sample period; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system I/O that a batch address space must be responsible for before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of batch address spaces to be listed whenever a batch I/O overload condition is detected. The maximum value allowed is 10.</td>
</tr>
<tr>
<td>rate</td>
<td>is the number of EXCPs-per-second that must be reached before exception messages are issued; the default is a system I/O rate greater than 25 EXCPs per second</td>
</tr>
</tbody>
</table>
**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the system I/O rate exceeds 50 EXCPs-per-second over the preceding two minutes:

**BIO 4,25,5,50**

A warning is issued for any batch address space doing more than 25 percent of the I/O.

**Warning messages**

The following warning message is issued by the BIO sampler:

PWSBIO95 *WARNING* I/O RATE FOR JOB nnnnnnnn IS xxx% OF SYSTEM TOTAL

Message PWSBIO95 indicates that the specified address space has exceeded the warning threshold for percentage of system I/O, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percent of the total system I/O rate</td>
</tr>
</tbody>
</table>

**BPAG**

The Batch Paging (BPAG) sampler produces a warning message when a batch address space uses more than a specified percentage of the total system paging activity during a reporting interval.

The message can be suppressed if a given threshold for the total system paging rate is not reached during the reporting interval.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length in intervals of the sample period; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system paging that a batch address space must be responsible for before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of batch address spaces to be listed whenever a batch paging overload condition is detected. The maximum value allowed is 10.</td>
</tr>
<tr>
<td>syspag</td>
<td>is the paging rate in pages-per-second that must be reached before exception messages are issued; the default is a system paging rate greater than 25 pages per second</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement produces a warning at two-minute intervals if the system paging rate exceeds 50 pages-per-second over the preceding two minutes:

**BPAG 4,25,5,50**

A warning is issued for any batch address space doing more than 25 percent of the paging.

Warning messages

The following warning messages are issued by the BPAG sampler:

**PWSBPAG0 *WARNING* TOTAL SYSTEM-PAGING RATE IS NOW ppp PAGES/SEC**

Message PWSBPAG0 indicates that the system paging rate has exceeded the warning threshold.

**PWSBPAG1 *WARNING* PAGING RATE FOR JOB nnnnnnnn IS xxx% OF TOTAL SYSTEM PAGING**
Message PWSBPAG1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percent of total system paging rate</td>
</tr>
</tbody>
</table>

BSRV

The Batch Service (BSRV) sampler produces a warning message when a batch address space uses more than the specified percentage of one or more SRM service measures during a reporting interval.

You can suppress the message if a given total system service-rate threshold is not reached during the reporting interval.

You can use multiple BSRV sampler statements; for example, you might want to have one TSRV statement for each measure.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

![BSRV Syntax Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length in intervals of the sample period; the default sample period is 30 seconds (one interval)</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| percent   | is the percentage of total system SRM component utilization for which an address space is responsible  
This total must be exceeded before an exception message is issued for that address space. |
| measure   | is the SRM service measure to be monitored:  
C indicates CPU service units  
I indicates I/O service units  
M indicates MSO service units  
T indicates total service units; the default |
| rptno     | is the maximum number of address spaces to be listed whenever a batch service overload condition is detected; the default is 5 |
| minrate   | is the system SRM service-consumption rate that must be reached before exception messages are issued; the default is a system service rate greater than zero SUs per second |

### Example

Assuming a report interval of 30 seconds, the following control statement displays up to five warning messages at two-minute intervals whenever an address space being monitored uses more than the 35 percent of total SRM components service during a given reporting interval:

**BSRV 4,35,t,5,509**

You can suppress the message if a given total system service rate threshold of 50 is not reached during a reporting interval.

### Warning messages

The following warning message is issued by the BSRV sampler:

**PWSBSRV1 *WARNING* mmm SERVICE RATE FOR JOB jjjjjjjjj IS ppp % OF nnn SUs**

Message PWSBSRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmm</td>
<td>is the SRM service component being monitored</td>
</tr>
<tr>
<td>jjjjjjjjj</td>
<td>is the address space name</td>
</tr>
<tr>
<td>ppp</td>
<td>is the address space SRM component use as a percentage of the system total</td>
</tr>
<tr>
<td>nnn</td>
<td>is the total number of service units</td>
</tr>
</tbody>
</table>
The Common Area Dataspace (CAD) Percentage sampler determines the percentage of the permitted Common Area Dataspaces that have been created and produces a warning message when the user-specified percentage has been exceeded.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals. The default value is 2. The default sample period of 30 seconds (one interval) is the percentage utilization of CSA that causes a warning message to be issued.</td>
</tr>
<tr>
<td>percent</td>
<td>is the common area dataspace percentage. When the CAD usage exceeds this percentage, a warning message to be issued. Specify a percentage between 0 and 99. The current usage percentage is calculated by dividing the number of CAD dataspaces in use by the maximum number of ALE slots in a PASN access list that are reserved for CADs. The result of that calculation is then compared to the threshold that you supply and, if the usage is greater than the supplied threshold, a warning message is generated.</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message at 30-second intervals whenever the CAD utilization exceeds 40 percent:

CADP 1,40

**Warning message**

The following warning message is issued by the CADP sampler:

PWSCADP1 *Warning* CAD utilization is percent% Used=used Lost=lost

Message PWSCADP1 indicates that the permitted Common Area Dataspaces has exceeded the user-specified percentage, where
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>percent</td>
<td>current percentage of CAD dataspace used</td>
</tr>
<tr>
<td>used</td>
<td>number CAD dataspace used</td>
</tr>
<tr>
<td>lost</td>
<td>number of CAD dataspace lost</td>
</tr>
</tbody>
</table>

**CHA**

The Channel (CHA) sampler monitors channel-path usage and produces a warning message when a channel path exceeds the defined utilization-percentage threshold.

**Syntax**

```
CHA int threshold chpid1 chpid2
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>threshold</td>
<td>is the average utilization high threshold (0-100 percent)</td>
</tr>
<tr>
<td>chpid1</td>
<td>is the lowest path number in the range of channel paths to be monitored; must be two digits</td>
</tr>
<tr>
<td>chpid2</td>
<td>is the highest path number in the range of channel paths to be monitored; must be two digits</td>
</tr>
</tbody>
</table>

**Note**

Parameters `chpid1` and `chpid2` are optional. If they are omitted, all channel paths are monitored.

**Example 1**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at one-minute intervals if any channel path in the range 01 through 10 exceeds 35 percent utilization:

```
CHA 2,35,01,10
```
Example 2

Assuming a report interval of 30 seconds, the following control statement displays warning messages at 30-second intervals if any channel path utilization exceeds 40 percent:

CHA 1,40

Warning message

The following warning message is issued by the CHA sampler:

PWSCHA01 *WARNING* CHANNEL xx ON CPU y UTILIZATION IS zz%

where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx</td>
<td>is the physical channel number (0 to 15)</td>
</tr>
<tr>
<td>y</td>
<td>is the CPU number (0 or 1)</td>
</tr>
<tr>
<td>zz</td>
<td>is the percent utilization</td>
</tr>
</tbody>
</table>

CPAG

The critical p aging (CPAG) sampler scans all address spaces each report interval to determine if pages have been stolen from any critical address spaces.

If so, a warning message is issued. A critical address space is an address space with bit RAXCriticalPaging on in its RAX control block. A warning message is issued if bit RaxCriticalPagesStolen is also on.

Syntax

The parameter is defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
</tbody>
</table>

**Example**

Assuming that the report interval is 30 seconds, the CPAG sampler executes once every 60 seconds and scans all address spaces. A warning message is issued for any critical address space from which pages have been stolen.

**CPU**

The CPU sampler can monitor CPU usage for the entire LPAR, TSO users (as a group), and CPU utilization for individual address spaces.

The CPU sampler provides a view of CPU utilization as observed by the LPAR. Use the PCPU sampler for a view of CPU utilization as observed by the sysplex.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length in intervals of the sample period; the default sample period is 30 seconds (one interval)</td>
</tr>
</tbody>
</table>
| CPULIMIT=nn       | is the average CPU utilization for all CPUs in an LPAR that causes a warning message to be issued Specify two digits; the valid range is 0 to 99.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSOLIMIT=nn</td>
<td>is the percent CPU utilization by TSO users that causes a warning message to be issued</td>
</tr>
<tr>
<td></td>
<td>Specify two digits to represent the percentage; the valid range is 0 to 99.</td>
</tr>
<tr>
<td>JOBLIMIT=nn</td>
<td>is the percent CPU utilization by individual jobs, TSO users, or Started Tasks that causes a warning message to be issued</td>
</tr>
<tr>
<td></td>
<td>Specify two digits to represent the percentage; the valid range is 0 to 99.</td>
</tr>
<tr>
<td>STCLIMIT=nn</td>
<td>is the percent CPU utilization by started tasks</td>
</tr>
<tr>
<td></td>
<td>The valid range is 0-99.</td>
</tr>
</tbody>
</table>

**Note**

If you omit a threshold specification or specify zero, no warning messages are produced.

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at five-minute intervals if CPU0 usage exceeds 99 percent, TSO usage exceeds 50 percent, or any job uses more than 20 percent of the CPU:

**CPU 10, TSOLIMIT=50, JOBLIMIT=20**

**Warning messages**

The following warning messages are issued by the CPU sampler:

**PWSCPU01** *WARNING* CPU usage is *percent*

Message PWSCPU01 indicates that the combined TSO, STC, and batch CPU usage exceeds the CPULIMIT threshold during the reporting interval, where *percent* is the percentage of CPU time that the system was busy.

**PWSCPU02** *WARNING* *type* CPU utilization is *percent*

Message PWSCPU02 indicates that the total CPU usage exceeds the warning threshold, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>type</em></td>
<td>is the type of user (TSO, JOB, STC)</td>
</tr>
<tr>
<td><em>percent</em></td>
<td>is the percentage of the current CPU utilization for the specified type</td>
</tr>
</tbody>
</table>
CQD

The CPU Queue Depth (CQD) sampler monitors the CPU active task queue and produces a warning message when the average queue depth of tasks that are ready to run and waiting for a processor exceeds a threshold you specify.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>depth</td>
<td>is the threshold number of tasks that are ready to execute and waiting for a processor</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at 30-second intervals if the average CPU queue depth exceeds 10:

```
CQD 1,10
```

Warning message

The following warning message is issued by the CQD sampler:

```
PWSCQD01 *WARNING* CPU QUEUE DEPTH IS xxx
```

Message PWSCQD01 indicates that xxx is the average number of tasks that are waiting for a processor.

CSA

The CSA and SQA Utilization (CSA) sampler monitors CSA usage and system-defined critical thresholds for SQA usage, and anticipates problems caused by CSA or SQA usage that exceeds the thresholds.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>csapc</td>
<td>is that a warning is produced when the amount of allocated CSA, as determined by the chosen method, reaches or exceeds the specified value</td>
</tr>
<tr>
<td>ecsapct</td>
<td>is that a warning is produced when the amount of allocated ECSA, as determined by the chosen method, reaches or exceeds the specified value</td>
</tr>
</tbody>
</table>
| method    | is how the fragmented free storage should be treated  
  There are two methods:  
  - ALOC indicates that the fragmented free storage should be treated as allocated  
    CSA pages that contain some free space are committed (allocated) to a specific subpool and key; therefore, the availability of that free space is limited.  
  - FREE indicates that the fragmented free storage should be treated as free (available for allocation)  
  If omitted, the method is ALOC. |

Example

Assuming a report interval of 30 seconds, the following control statement displays a warning message at 30-second intervals whenever CSA or ECSA utilization exceeds 80 percent:

```
CSA 1,80,80,FREE
```

Warning messages

The following warning messages are issued by the CSA sampler:

```
WSCSA01 *WARNING* CSA usage is xxx %; remaining free space is yyyyyyyy
```

Message PWSCSA01 indicates that CSA usage exceeds the specified threshold, where
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| xxx      | is the percentage of CSA used  
The percentage used includes fragmented free storage. |
| yyyyyyy  | represents the amount of CSA still available |

**PWSCSA02 *WARNING* SQA CRITICAL THRESHOLD EXCEEDED**

Message PWSCSA02 indicates that SQA utilization has exceeded a critical level.

**PWSCSA03 *WARNING* SQA APPROACHING CRITICAL LEVEL**

Message PWSCSA03 indicates that SQA utilization is reaching a critical level.

**PWSCSA90 SPECIFIED THRESHOLD GREATER THAN 100%**

Message PWSCSA90 indicates that the CSA control statement was rejected because the specified threshold is invalid.

---

**CSMJ**

The CSMJ sampler monitors the percentage of common storage use by each address space and produces a warning for each one that meets or exceeds the specified thresholds.

This information allows you to take action before a common storage shortage adversely affects the entire system. The thresholds represent the percent of total available common storage (CSA, ECSA, SQA, or ESQA) that has been allocated by any single job name.

---

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

```plaintext
CSMJ 1
  csa  ecsa  sqa  esqa
```

The parameters are defined as follows:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>csa</td>
<td>is the percent allocation of CSA by any job name that causes a warning message to be issued</td>
</tr>
<tr>
<td>ecsa</td>
<td>is the percent allocation of ECSA by any job name that causes a warning message to be issued</td>
</tr>
<tr>
<td>sqa</td>
<td>is the percent allocation of SQA by any job name that causes a warning message to be issued</td>
</tr>
<tr>
<td>esqa</td>
<td>is the percent allocation of ESQA by any job name that causes a warning message to be issued</td>
</tr>
</tbody>
</table>

**Warning messages**

The following warning messages are issued by the CSMJ sampler:

```
PWSCSM01 *WARNING* xxxxxxxxx yyyy HAS ALLOCATED nn% OF zzzz
```

Message PWSCSM01 indicates that the percentage of a common storage area allocated by a job exceeds the specified threshold, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxxx</td>
<td>is the job name</td>
</tr>
<tr>
<td>yyyy</td>
<td>is the Address Space Identifier (ASID)</td>
</tr>
<tr>
<td>nn</td>
<td>is the percentage of the common storage area allocated by the job</td>
</tr>
<tr>
<td>zzzz</td>
<td>is the common storage area (CSA, ECSA, SQA, or ESQA)</td>
</tr>
</tbody>
</table>

**DEV**

The Device Busy (DEV) sampler monitors the amount of activity on either a single device or a range of device addresses and identifies excessive contention for devices.

**Syntax**

```
DEV [int] threshold [dvn1, dvn2] [dvn]
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>threshold</td>
<td>is the threshold percentage for device utilization A warning message is issued for each device in the range whose percent busy exceeds this threshold.</td>
</tr>
<tr>
<td>dvn</td>
<td>is the hexadecimal address of a single device to be monitored The device number does not have to be assigned to a specific device at the time you invoke the sampler. When the device number is assigned to a specific device, monitoring begins at that point.</td>
</tr>
<tr>
<td>dvn1</td>
<td>is the first hexadecimal device address of the range of devices to be monitored</td>
</tr>
<tr>
<td>dvn2</td>
<td>is the last hexadecimal device address of the range of devices to be monitored</td>
</tr>
</tbody>
</table>

**Note**
You can have up to 16 DEV samplers, each with different thresholds, active at one time, just as long as the device numbers do not overlap.

**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message at five-minute intervals when any device in range 150 through 154 is more than 20 percent busy over the five-minute interval:

```
DEV 10,20,150,154
```

A separate message is issued for each device that exceeds the 20 percent threshold.

**Warning message**

The following warning message is issued by the DEV sampler:

```
PWSDEV01 *WARNING* DEVICE dvn (vvvvvv) USAGE IS xxx PERCENT
```

Message PWSDEV01 indicates that the utilization of device dvn exceeds the warning threshold. For disk and tape devices, the volume serial is also displayed (vvvvvv).

**DIOQ**

The DASD I/O Queue (DIOQ) sampler displays information for a DASD device when the number of I/Os queued to the device equals or exceeds the specified queue depth.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the number of intervals between report periods; the default report period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>depth</td>
<td>is the queue depth necessary for the device to be reported For example, if you specify 3, all DASD devices with three or more queued requests are reported. The minimum value is 1 and the maximum is 999; the default is 5.</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the I/O Queue depth is greater than or equal to 3:

```
DIOQ 4,3
```

Warning message

The following message is issued by the DIOQ sampler:

```
PWS DioQ0 *WARNING* Q=qqq nnnn aaa volser s mmm jobname
```

Message PWS DioQ0 indicates that the I/O Queue depth has exceeded the warning threshold, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qqq</td>
<td>is the I/O queue depth for the DASD device</td>
</tr>
<tr>
<td>nnnn</td>
<td>is the device type</td>
</tr>
<tr>
<td>aaa</td>
<td>is the device address</td>
</tr>
<tr>
<td>volser</td>
<td>is the volume serial number</td>
</tr>
<tr>
<td>s</td>
<td>is the special status indicator The letter P indicates that the device contains an active paging data set. The letter R indicates that the device is currently reserved.</td>
</tr>
<tr>
<td>mmm</td>
<td>is the mount attribute for the volume</td>
</tr>
</tbody>
</table>
Variable Description

jobname is the job name for the job with I/O in progress at the time that the device was checked

DNR

The Device Not Ready (DNR) sampler monitors online tape and DASD devices that are not ready and produces a warning message if they have not been ready for more than a specified interval.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>notready</td>
<td>is the number of minutes a device must remain not ready for a warning message to be issued</td>
</tr>
</tbody>
</table>

You can exempt the reporting of specific devices by specifying them in BBPARM member $$XDEV. For more information, see “$$XDEV” on page 30.

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at one-minute intervals for each online device that has been not ready for over four minutes:

DNR 2,4

Warning message

The following warning message is issued by the DNR sampler:

PWSDNRO1 *WARNING* DEVICE nnn (vvvvvv) NOT READY FOR xx MIN
Message PWSDNR01 indicates that device nnn has been online but not ready for a period of time that exceeds the warning threshold.

**Note**
A blank volume serial number appears when volume rollover occurs.

**DVTN**

The DVTN sampler checks the list of job names and the expected operational time range that you specified.

The DEV service also displays a message for each job that is not executing during the specified time range. When the ACT=command parameter is specified and the job is not executing during the assigned timeframe, the samplr executes the MVS command.

The DVTN service displays a message for each job that is not executing during the specified time range. When the ACT=command parameter is specified and the job is not executing during the assigned timeframe, the samplr executes the MVS command.

The DVTN sample obtains the list of jobs from one of the following places:

- the logical PARMLIB concatenation
- the data set concatenation that is used to obtain the Exception Monitor parameters (see “DEVIATN” on page 87)

The DVTN service retains the location for the job list and uses the location on subsequent executions of the sampler, unless you specify a new value. The member name must begin with CONFIGxx followed by a user-specified one or two-alphanumeric suffix. The default suffix is 00. For more information, see “DEVIATN” on page 87.

**Syntax**

```
DVNT
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 20 intervals</td>
</tr>
</tbody>
</table>
### Syntax for the CONFIGxx member

The entries in the CONFIGxx member are entered as comment statements so that they can be included in the IBM PARMLIB member CONFIGxx.

The statements can also reside in other data sets. Each entry must use the following format:

`*/NAME=identifier TIME=hhmm-hhmm [ACT=command]`

The statement must begin in column one of the data set. Separate the keyword parameters and their values with spaces. The identifier and time range values are required and the ACT=command parameter is optional.

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>job name (or Started Task ID for Started Tasks) of the job to be monitored</td>
</tr>
</tbody>
</table>

You can specify one of the following values:

- `jobname`
- `jobname.stid`
- `.stid`

For example, if the job name is ABC and the started task ID is DEF then you could specify `ABC`, `ABC.DEF` or `.DEF`.

If the job name and the started task ID are not unique names (in other words, they are also used by other address spaces), you must specify both the job name and started task ID.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| hhmm-hhmm | beginning and ending values for the interval when the specified job must be active where:  
- \( hh \) represents the hour (represented as a twenty four-hour clock value)  
- \( mm \) represents the minute within the hour  
The first \( hhmm \) value is the beginning of the range and the second \( hhmm \) entry represents the end of the range. Both values are required. The beginning value must be a number that is less than the ending value number. A beginning value of zero indicates 24-hour operation. |
| ACT=command | is an optional field that you can substitute with any MVS operator command  
The MVS operator command is executed when the specified job is not executing during the specified time interval. A commonly used MVS operator command is the START command. |

### Example

In this example, the DVTN sampler control statement is as follows:

**DVTN 20,DP**

If the report interval is 30 seconds (the default), the DVTN sampler executes every ten minutes (20 x 30 seconds). In addition, this example assumes the following conditions:

- the DVTN sampler executes at 23:05
- the started task DAYEND is not executing
- the CONFIGDP member contains the following statements:

  — /*NAME=ABC.DEF TIME=1000-2200
  — /*NAME=DAYEND TIME=2300-2310,ACT=S DAYEND

In this example, the DVTN sampler issues the following message and the MVS operator command **S DAYEND** is executed:

PWSCF8I WARNING DAYEND should be active 2300-2315

### ENQ

The Enqueue Conflicts (ENQ) sampler monitors enqueue conflicts and produces a warning message when jobs have been waiting for enqueued resources for more than a specified interval.
You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 23.

You can suppress reporting for specific major and minor enqueue names by including them in BBPARM member $XENQ. See “$XENQ” on page 32.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 20 sample intervals (ten minutes when the report interval is 30 seconds, the default)</td>
</tr>
<tr>
<td>waiting</td>
<td>is the number of minutes that an enqueue conflict must exist for a warning message to be issued. The default value is 10 minutes.</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if any enqueue conflicts have existed for over five minutes:

ENQ 4,5

Warning message

The following warning message is issued by the ENQ sampler:

PWSENQ01 *WARNING* ENQ conflict over n Min (qqqqqqqqq) w waiter(s)

Message PWSENQ01 indicates that one or more jobs have been waiting for resources for longer than the specified threshold. To display the enqueue situation, use service ENQUEUES.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>is the number of minutes the ENQ conflict has existed</td>
</tr>
</tbody>
</table>
### FIX

The Fixed Pages (FIX) sampler monitors the amount of page-fixed common storage by its location in real memory.

A warning message is issued when a threshold value is exceeded.

The monitor totals the fixed storage into one of the following groups:

**Table 9: Real storage locations monitored by the FIX sampler**

<table>
<thead>
<tr>
<th>Storage type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all real storage</td>
<td>All real storage, regardless of its location</td>
</tr>
<tr>
<td>below-the-line</td>
<td>storage backed by real memory with addresses less than 16MB</td>
</tr>
<tr>
<td>above-the-line and below-the-bar</td>
<td>storage backed by real memory with addresses between 16M to 2G</td>
</tr>
<tr>
<td>above-the-bar</td>
<td>storage backed by real memory with addresses above 2G</td>
</tr>
</tbody>
</table>

### Syntax

```
FIX int,a,b,c,d
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int</code></td>
<td>is the number of report intervals between warning messages; default interval is 30 seconds</td>
</tr>
<tr>
<td><code>a</code></td>
<td>is the threshold for the number of real memory pages used to back page-fixed common storage that, when exceeded, produces a warning message</td>
</tr>
<tr>
<td></td>
<td>Note: This is the sum of the values that is reported by <code>b</code>, <code>c</code>, and <code>d</code> (below) when those messages are displayed.</td>
</tr>
<tr>
<td><code>b</code></td>
<td>is the threshold for the number of real memory pages located above the 16M line, and below the bar used to back page-fixed common storage that, when exceeded, produces a warning message</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
c | is the threshold for the number of real memory pages located above the 16M line, and below the bar used to back page-fixed common storage that, when exceeded, produces a warning message
d | is the threshold for the number of real memory pages located above the bar used to back page-fixed common storage that, when exceeded, will produce a warning message

**Note**
Specifying a value of zero for \( a, b, c, \) or \( d \) will suppress that warning message.

**Example**

**FIX 2,0,32,128,0**

Warning messages will be issued once a minute (assuming the default interval of 30 seconds is in effect). A warning message will be issued if the amount of fixed common storage backed below-the-line exceeds 32 pages, or the amount of fixed common storage backed above-the-line but below-the-bar exceeds 128 pages. Fixed common storage backed above-the-bar is not monitored.

**Messages**

PWSFIX01 *WARNING* Page-fixed memory= x pages

PWSFIX02 *WARNING* Below-the-Line page-fixed memory= x pages

PWSFIX03 *WARNING* Above-the-Line page-fixed memory= x pages

PWSFIX04 *WARNING* Above-the-Bar Page-fixed memory= x pages

**INT**

The Interval (INT) sampler changes the reporting base interval.

For each sampler, you can specify a multiple of the report base interval in the interval parameter of the control statement.

**Syntax**

```
INT reportInterval
```
The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| reportInterval | is the interval between report phases in seconds  
                  The range is from 5 to 60 seconds; the default is 30 seconds. |

**Example**

To set a report interval of 60 seconds, place this control statement in BBPARM library member PWSCPM xx:

```
INT 60
```

The base interval between report phases is 60 seconds.

**JET**

The Job Elapsed Time (JET) sampler monitors the elapsed time of jobs and produces a warning message if any job exceeds a user-specified interval.

The threshold interval is specified by job class. The sampler can also be used to enforce installation standards relating job class to duration.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
</tbody>
</table>
| class     | is the job class to be monitored (A through Z and 0 through 9)  
                  You can also specify an asterisk (*) to indicate any class. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>is the threshold elapsed time in minutes</td>
</tr>
<tr>
<td></td>
<td>A value of 0 indicates that no warnings will be issued for the class.</td>
</tr>
</tbody>
</table>

**Note**
You can specify up to 15 pairs of job class and time.

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals for every job running in Class A for longer than five minutes and any other class for longer than ten minutes:

JET 4,A,5,*,10

**Note**
JES3 job classes are not supported.

**Warning message**

The following warning message is issued by the JET sampler:

PWSJET02 *WARNING* Class H Job nnnnnn jjjjjjjj sssssssss executing xxx minutes

where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>is the job class</td>
</tr>
<tr>
<td>nnnnnnn</td>
<td>is the JES2 job number</td>
</tr>
<tr>
<td>jjjjjjj</td>
<td>is the job name</td>
</tr>
<tr>
<td>sssssssss</td>
<td>is the service class name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the elapsed time in minutes</td>
</tr>
</tbody>
</table>

**JSU**

The Job Storage Usage—Real & Auxiliary (JSU) sampler monitors all jobs’ usage of ASM slots and real frames, and identifies jobs that are overusing real storage and allocated slots of auxiliary storage.
Note
You can suppress reporting for specific address spaces by including their names in
member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| int | is the length of the sample period in intervals  
The default sample period is 30 seconds (one interval). |
| frames | is the maximum number of frames that can be held by a job (real storage) before a message is issued  
You can specify up to 9999999. |
| slots | is the maximum number of slots that can be held by a single job (auxiliary storage) before a message is issued  
You can specify up to 9999999. |

Note
If you omit threshold specifications or specify zero for all threshold values, no warning message is issued.

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages every five minutes for each job that holds more than

- 100 frames
- 200 slots

JSU 10,100,200

Warning messages

The following warning messages are issued by the JSU sampler:

PWSJSU01 *WARNING* ttt jjjjjjjj HOLDs nnnn FRAMES
The PWSJSU01 message indicates that a job has exceeded the number of frames that were specified in the JSU sampler command, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is the type, one of JOB, TSU, or STC</td>
</tr>
<tr>
<td>jjjjjjjjj</td>
<td>is the job name of the job exceeding the threshold</td>
</tr>
<tr>
<td>nnnn</td>
<td>is the count of the number of frames held by this job</td>
</tr>
</tbody>
</table>

PWSJSU02 *WARNING* ttt jjjjjjjj HOLDS nnnn slots

The PWSJSU02 message indicates that a job has exceeded the number of slots that were specified in the JSU sampler command, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is the type, one of JOB, TSU, or STC</td>
</tr>
<tr>
<td>jjjjjjjjj</td>
<td>is the job name of the job exceeding the threshold</td>
</tr>
<tr>
<td>nnnn</td>
<td>is the count of the number of slots held by this job</td>
</tr>
</tbody>
</table>

JVR

The Job Awaiting V=R Region (JVR) sampler monitors jobs awaiting a V=R region and produces a warning message if any job is waiting for a V=R region.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

```
  JVR
    int
```

The parameter is defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
</tbody>
</table>
Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at five-minute intervals for each job waiting for a V=R region:

**JVR 10**

Warning messages

The following warning messages are issued by the JVR sampler:

**PWSJVR01 *WARNING* JOB jjjjjjjjjj WAITING FOR V=R REGION**

The PWSJVR01 message indicates that a job is waiting for V=R region, where jjjjjjjj is the job name of the waiting job.

MTP

The Mounts Pending (MTP) sampler monitors tape and DASD-mount requests, and produces a warning message when outstanding mount requests have been pending for more than a specified interval.

Syntax

```
MTP 4,10
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>pending</td>
<td>is the number of minutes that a mount must remain pending for a warning message to be issued</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals for each device with a mount pending for over 10 minutes:

**MTP 4,10**
Warning message

The following warning message is issued by the MTP sampler:

PWSMTP01 *WARNING* MOUNT FOR dvn (vvvvvv) OUTSTANDING FOR xxx MINUTES

Message PWSMTP01 indicates that the mount request for device dvn has been outstanding for a period of time that exceeds the warning threshold.

**Note**

A blank volume serial number (vvvvvv) appears when volume rollover occurs.

NRQ

The Non Reusable Queue (NRQ) sampler monitors the number of available address space vector table slots (RSVNONR) that are available for replacing address space IDs marked as non-reusable.

It produces a warning message when both of the following conditions occur:

- The number of available address space vector table slots (RSVNONR) that are available for replacing address space IDs marked as non-reusable drops below the specified threshold value (limit1).
- The number of slots that are available for starting new address spaces (available queue) drops below the specified threshold value (limit2).

**Syntax**

```
NRQ int limit1 limit2
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default interval is 30 seconds</td>
</tr>
<tr>
<td>limit1</td>
<td>is the threshold for slots reserved for replacing slots marked non-reusable. The number of slots on the non-reusable replacement queue (value in ASVTANR) must be less than or equal to the specified number for a warning message to be issued.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>limit2</td>
<td>is the threshold for the available queue (number of additional address spaces that can be started) The number of slots on the available queue (ASVTAAV) must be less than or equal to the specified number for a warning message to be issued.</td>
</tr>
</tbody>
</table>

**Note**

Slots (ASIDs) are marked non-reusable when the job terminates while in a cross-memory environment. They remain unusable until all address spaces that had binds with the address space have ended. The SYSPROG ASVT service displays the current and original values for the non-reusable replacement queue and the available queue.

### Example

To display a warning message for a specified threshold, type

NRQ 2,5,10

This NRQ sampler checks every two intervals to see if there are less than six (limit1 value) slots remaining on the non-reusable replacement queue and less than 11 (limit2 value) slots on the available queue. A warning message is issued if both limit1 and limit2 have been reached. The message is reissued every minute (two intervals), assuming that the default interval value is 30 seconds, unless the number of slots on the available queue exceeds the limit2 value.

### Warning message

The following warning message is issued by the NRQ service:

PWSNRQ01 *WARNING* Non-reusable replacement queue length is nnn

**NRQP**

The Non Reusable Queue Percentage (NRQP) sampler monitors the percentage of available address space vector table slots (RSVNONR) reserved for replacing address space IDs marked as non-reusable.

It produces a warning message when *both* of the following conditions occur:

- The percentage of available address space vector table slots (of RSVNONR) reserved for replacing address space IDs marked as non-reusable drops below the specified threshold value (limit1)
The percentage of slots available for starting new address spaces (available queue) drops below the specified threshold value \((\text{limit2})\).

**Syntax**

\[
\text{NRQP } \text{int } \text{limit1 } \text{limit2}
\]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{int}</td>
<td>is the length of the sample period in intervals; the default interval is 30 seconds</td>
</tr>
<tr>
<td>\text{limit1}</td>
<td>is the threshold for slots reserved for replacing slots marked non-reusable. The percentage of slots on the non-reusable replacement queue ((\text{ASVTANR} / \text{ASVTNONR} \times 100)) must be less than or equal to the specified percentage for a warning message to be issued.</td>
</tr>
<tr>
<td>\text{limit2}</td>
<td>is the threshold for the percentage of address spaces (\text{MAXUSER}) that can still be started. The percentage of the original number of slots on the available queue that are still available ((\text{ASVTAAV} / \text{ASVTMAXI} \times 100)) must be less than or equal to the specified percentage for a warning message to be issued.</td>
</tr>
</tbody>
</table>

**Note**

Slots (ASIDs) are marked non-reusable when the job terminates while in a cross-memory environment. They remain unusable until all address spaces that had binds with the address space have ended.

The SYSPROG ASVT service displays the current and original values for the non-reusable replacement queue and the available queue.

**Example**

To display a warning message for a specified percentage, type

\text{NRQP 10,25,10}

Every 10 intervals, this NRQP sampler checks to see if less than 26\% \((\text{limit1})\) value) of the slots remain on the non-reusable replacement queue. It also checks to see if less than 10\% \((\text{limit2})\) value) of \text{MAXUSER} (obtained from \text{ASVTMAXI}) slots remain on the available queue.

A warning message is issued if both \text{limit1} and \text{limit2} have been reached. The message is reissued every minute (two intervals), assuming that the default interval value is 30 seconds, unless the percentage of slots on the available queue exceeds the \text{limit2} value.
Warning message

The following warning message is issued by the NRQP service:

PWSNRQP1 *WARNING* Non-reusable replacement queue length is nnn percent of vvv

OUT

The Jobs Swapped Out (OUT) sampler monitors jobs, TSO sessions, and Started Tasks and produces a warning message for any that have been swapped out of memory for an interval that exceeds the user-specified threshold.

Note

TSO users in input or output terminal wait state are not considered.
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For more information, see “$$INEXCL” on page 23.

Syntax

![Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>time</td>
<td>is the threshold for swap-out in minutes</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at one-minute intervals for jobs, TSO users, or Started Tasks swapped out longer than one minute:

OUT 2,1
Warning message

The following warning message is issued by the OUT sampler:

PSWOUT01 *WARNING* ttt jjjjjjjj (nnnn) SWAPPED OUT xxxxx MINS FOR ** reason **

where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>address space type: JOB, STC, or TSU</td>
</tr>
<tr>
<td>jjjjjjjj</td>
<td>job name</td>
</tr>
<tr>
<td>nnnn</td>
<td>ASID</td>
</tr>
<tr>
<td>xxxxx</td>
<td>minutes swapped out</td>
</tr>
<tr>
<td>reason</td>
<td>swap reason</td>
</tr>
</tbody>
</table>

The swap reason will be one of the following:

- Long wait - LONG WAIT
- Auxiliary storage - AUX. STORAGE SHORTAGE
- Real storage - REAL STORAGE SHORTAGE
- Detected wait - MSO DETECTED WAIT
- Requested swap - REQUEST SWAP OUT
- Enqueue - ENQUEUE EXCHANGE
- Exchange - REC. VALUES EXCHANGE
- Unilateral - UNILATERAL SWAP OUT
- Transition - TRANSITION SWAP OUT
- Logical - LOGICAL SWAP OUT
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>rate</td>
<td>is the paging rate per second measured over the reporting period that triggers a warning message</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the system paging rate exceeds 30 pages-per-second over the preceding two minutes:

PAG 4,30

Warning message

The following warning message is issued by the PAG sampler:

PWSPAG01 *WARNING* PAGING RATE IS xxx PER SECOND

Message PWSPAG01 indicates that the paging rate has exceeded the warning threshold. The variable xxx is the paging rate per second measured over one reporting interval causing this warning to be triggered.

PCPU

The PCPU sampler monitors CPU utilization for the host partition relative to its relative-share value and produces a warning message when utilization is outside of the limits that you specified.

The limit values are specified as a percentage of the partition's relative-share. Plex overhead is excluded when computing the partition's CPU utilization.
A partition can receive more than 100% of its relative share.

**Syntax**

![Diagram of PCPU sampler with parameters int, lowpct, highpct, plexpct]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>int</strong></td>
<td>is the length of the sample period in intervals; the default is two intervals (60 seconds)</td>
</tr>
<tr>
<td><strong>lowpct</strong></td>
<td>is the low threshold percent</td>
</tr>
<tr>
<td></td>
<td>A warning message is issued if the partition receives less than this percentage of its relative-share of CPU over the sample period and the total Plex CPU utilization is at or above plexpct. A value of zero indicates that no lower limit checking is to be performed. The default is 0.</td>
</tr>
<tr>
<td><strong>highpct</strong></td>
<td>is the high threshold percent</td>
</tr>
<tr>
<td></td>
<td>A warning message is issued if the partition receives more than this percentage of its relative-share of CPU over the sample period and the total Plex CPU utilization is at or above plexpct. A value of zero indicates that no upper limit checking is to be performed. The default is 0.</td>
</tr>
<tr>
<td><strong>plexpct</strong></td>
<td>is the Plex minimum percent</td>
</tr>
<tr>
<td></td>
<td>Warning messages for the lowpct and highpct conditions are suppressed when the plexpct value is less than the specified value (unless a value of 0 is specified). The default value is 80.</td>
</tr>
</tbody>
</table>

**Example 1**

PCPU,2,80,200,90

The reporting period is two intervals (one minute if the default interval is used). A warning message will be issued if the Plex is at least 90% utilized during the period and the host partition receives either less than 80% or more than 200% of its relative-share.

**Example 2**

PCPU,1,75,0,80
The reporting period is one interval (30 seconds if the default interval is used). A warning message will be issued if the Plex is at least 80% utilized (the default value) during the period and the host partition receives less than 75% of its relative-share. No upper limit checking is performed.

**Warning messages**

The following warning messages are issued by the PCPU sampler:

**PWSPCPU1** *WARNING* Partition xxxxxxxx received nn.nn% of its Relative Share

Message PWSPCPU1 is issued when the partition receives less than the lowpct value during the period.

**PWSPCPU2** *WARNING* Partition xxxxxxxx received nn.nn% of its Relative Share

Message PWSPCPU2 is issued when the partition receives more than the highpct value during the period.

**PER**

The PER sampler monitors the percentage of CPU resources that the system is using to process an active PER SLIP trap.

Because an active PER trap can cause serious performance degradation, it is important that you be notified when a PER trap is active using excessive CPU resources.

**Syntax**

![PER Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>pct</td>
<td>is the percentage of resource utilization required to produce a warning message. The default is 0. A warning message is produced when this threshold is reached or exceeded.</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if PER trap utilization exceeds 10% over the preceding two minutes:

**PER 4,10**

**Warning message**

The following warning message is issued by the PER sampler when a PER SLIP trap is active and its resource utilization exceeds the threshold:

```
PWSPER01 *WARNING* PER SLIP trap xxxx active yy % of the time in the last zzzzzz seconds
```

where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxx</td>
<td>is the four-character SLIP trap ID specified when the SLIP was set</td>
</tr>
<tr>
<td>yy%</td>
<td>shows the percentage of samples taken in which the PER trap was found to be active and indicates the percentage of time that the system performance was impacted</td>
</tr>
<tr>
<td>zzzzzz</td>
<td>gives the time over which the samples were taken. This time is the Exception Monitor reporting interval in seconds times the interval field from the PER control card in your PWSCPMxx member.</td>
</tr>
</tbody>
</table>

**PGQ**

The Page Data Set I/O Queue (PGQ) sampler monitors the depth of I/Os queued to page data sets and produces a warning if the average I/O queue depth to any page data set exceeds a specified threshold.

**Note**

Average queue depth computation includes samples that are taken when queue depth is zero; therefore, it can be less than one.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>depth</td>
<td>is the threshold average of I/Os queued to any page or swap data set</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the average count of I/Os queued to any page-data set exceeds five:

```
PGQ 4,5
```

Warning message

The following warning message is issued by the PGQ sampler:

```
PWSPGQ03 *WARNING* I/O QUEUE DEPTH FOR PAGE DATA SET nn ON vvvvvv IS xx
```

where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>is the data set number</td>
</tr>
<tr>
<td>vvvvvv</td>
<td>is the volume serial of device</td>
</tr>
<tr>
<td>xx</td>
<td>is the I/O queue depth</td>
</tr>
</tbody>
</table>

PVT

The private storage (PVT) sampler scans all address spaces each report interval to determine if any are exceeding the utilization threshold for below-the-line or above-the-line private storage.
If so, a warning message is issued.

In addition to the two threshold values specified on the PVT sampler control statement, which applies to all address spaces, you can set thresholds for individual address spaces in BBPARM member $$PVT. For more information, see “$$PVT” on page 28.

The PVT sampler treats fragmented free space as allocated storage, which is the more conservative approach since. The fragmented free space is still potentially available for use but its use is limited by the assigned subpool, key, and by size.

BMC recommends using a reporting period of five minutes or more because this sampler can generate a relatively high amount of overhead.

**Syntax**

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| int       | is the length of the sample period in intervals; the default sample period is 5 minutes (ten intervals)  
BMC recommends setting an interval of at least 5 minutes (10 intervals). |
| below     | is the below-the-line percentage threshold  
A warning message is displayed if the amount of below-the-line allocated storage equals or exceeds this value. Valid values can be from 0 to 100. A value of 100 indicates that no checking is performed. The default value is 90. |
| above     | is the above-the-line percentage threshold  
A warning message is displayed if the amount of above-the-line allocated storage equals or exceeds this value. Valid values can be from 0 to 100. A value of 100 indicates that no checking is performed. The default value is 80. |
```

**Example**

A warning message is issued for all address spaces that have allocated 80 percent or more of the below-the-line private storage, or 50 percent or more of the above-the-line private storage, unless other threshold values were specified in BBPARM member $$PVT. The example sampler will execute once every five minutes.

PVT 10,80,50
The Replies Outstanding (REP) sampler monitors messages to the operator (WTORs) that require replies.

This sampler produces a warning when replies are outstanding over a user-specified interval. This sampler also lists the outstanding replies.

Note
You can exclude messages that are issued by specific address spaces (see “$SRPJOBS” on page 29) or that contain specific text (see “$$RPTEXT” on page 31).

Syntax

```
REP int minutes
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int</code></td>
<td>is the length of the sample period in intervals and the time between consecutive samples for outstanding replies; the default is 30 seconds (one interval)</td>
</tr>
<tr>
<td><code>minutes</code></td>
<td>is the minimum amount of time, in minutes, that a reply must be outstanding before it is reported</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The actual time elapsed before an outstanding reply is reported might often be less than the value entered because a reply that is outstanding during a sampling interval is assumed outstanding for the entire interval. To improve accuracy use a small value for <code>int</code>.</td>
<td></td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at one-minute intervals if a reply has been outstanding for more than five minutes:

```
REP 2,5
```

Warning message

The following warning message is issued by the REP sampler:

```
PWSREPO1 *WARNING* REPLIES OUTSTANDING FOR OVER nnnnnnn MINUTES
```
Message PWSREP01 indicates that replies were outstanding for *nnnnnn* minutes.

**RES**

The DASD Reserves (RES) sampler monitors the reserved status of devices in a shared DASD system.

This sampler produces a warning when devices are reserved for more than a specified interval.

*Note*

- You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.
- To prevent reserve information for specific volumes from displaying, add the VOLSER to BBPARM member $$XRES. See “$$XRES” on page 33.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>int</em></td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td><em>minutes</em></td>
<td>is the number of minutes that a device must be reserved for a warning message to be issued</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at five-minute intervals if any device has been reserved for over 10 minutes:

**RES 10,10**
**Warning message**

The following warning message is issued by the RES sampler:

PWSRES01 *WARNING* DEVICE ddd (vvvvvv) RESERVED FOR mm,mmm MIN

Message PWSRES01 indicates that the device on ddd with volume serial number vvvvvv has been reserved for a period of time (mm,mmm) that exceeds the warning threshold.

**RSU**

The Real Storage Usage (RSU) sampler monitors the status of real storage within the system.

This sampler produces a warning for address spaces that consume excessive amounts of real storage when the low SYSEVENT has been issued.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of address spaces to be listed whenever low SYSEVENT has been issued; the default is 5, and the maximum value that you can specify is 64</td>
</tr>
</tbody>
</table>
Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages at one-minute intervals whenever the low SYSEVENT has been issued for the address spaces of the top seven real storage users:

**RSU 2,7**

Warning message

The following warning message is issued by the RSU sampler:

```
PWSRSU02 *WARNING* REAL STORAGE USE FOR ttt uuuuuuuu IS 12345678 FRAMES
```

Message PWSRSU02 indicates that the low SYSEVENT has been issued, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttt</td>
<td>is STC, TSU, or JOB</td>
</tr>
<tr>
<td>jjjjjjjjj</td>
<td>is the address space name</td>
</tr>
<tr>
<td>12345678</td>
<td>is the number of real-storage frames</td>
</tr>
</tbody>
</table>

SMF

The SMF sampler produces a warning if an SMF data set is full or SMF recording is not active, helping to ensure that no SMF data is lost.

Syntax

```
  SMF
   /
  int  pct
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>pct</td>
<td>is the percentage of space in all SMF data sets that has been filled</td>
</tr>
</tbody>
</table>
Example

Assuming a report interval of 30 seconds, the following control statement displays a warning message every five minutes if the SMF data sets are at least 80 percent full:

```
SMF 10,80
```

Warning messages

The following warning messages are issued by the SMF sampler:

**PWSSMF01** *WARNING* SMF IS NOT RECORDING

Message PWSSMF01 indicates that SMF data is not being collected. Accounting information will be lost.

**PWSSMF02** *WARNING* datasetname is full datasetname is full

Message PWSSMF02 indicates that the indicated SMF data set is full. The SMF data set should be dumped as soon as possible so that SMF can reuse it.

**PWSSMF03** *WARNING* All SMF data sets are full; xxxx records lost

Message PWSSMF03 indicates that SMF is unable to write records because both data sets are full; xxxx represents the number of records that SMF could not write.

**PWSSMF04** *WARNING* datasetname is partially full and inactive

Message PWSSMF04 indicates that the SMF data set is partially full and inactive, and should be dumped as soon as possible so SMF can reuse it.

**PWSSMF09** *WARNING* SMF data sets are xxx% full

Message PWSSMF09 indicates that the SMF data sets have used over a threshold percentage amount of their allocated space. You should monitor the SMF data set usage and dump them when they are filled.

**SPA**

The DASD Free Space (SPA) sampler monitors free space on specified DASD volumes.

**Note**

The SPA sampler does not support the extended area of Extended Addressability Volume (EAV) devices.
This sampler produces a warning if the free space on any specified DASD volume falls below the specified number of cylinders.

**Syntax**

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>cyl</td>
<td>is the low threshold of free cylinders</td>
</tr>
<tr>
<td>volser</td>
<td>is the volume serial number; the maximum number of volsers you can specify on a single statement is seven The volsers that you specify must exist at the time you invoke the sampler.</td>
</tr>
<tr>
<td>dvn</td>
<td>is the number of the device to be monitored; the maximum number of devices you can specify on a single statement is 14</td>
</tr>
</tbody>
</table>
```

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at five-minute intervals if free space on WORK01, WORK02, or WORK03, or on any volume mounted on device address 380 or 381, falls below 50 cylinders:

```
SPA 10,50,WORK01,WORK02,WORK03,380,381
```

**Warning message**

The following warning message is issued by the SPA sampler:

```
PWSSPA01 *WARNING* dvn vvvvvv has xxx free cylinders and yyyy free tracks
```

where

```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvn</td>
<td>is the device address</td>
</tr>
<tr>
<td>vvvvvv</td>
<td>is the volume serial of device</td>
</tr>
</tbody>
</table>
```
Variable | Description
---|---
xxx | is the count of free cylinders
yyyy | is the count of free tracks (not including cylinders)

## SPLW

The SPLW sampler produces a warning for address spaces that are waiting for SPOOL space.

### Syntax

```
SPLW int max
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the report period in intervals; the default report period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>max</td>
<td>is the maximum number of address spaces that will be reported during one report period The default is 20; the maximum permitted value is 99.</td>
</tr>
</tbody>
</table>

### Example

**SPLW 4,15**

The reporting period is four intervals (two minutes if the default interval is used). A warning message will be issued for each of the first 15 address spaces that are waiting for SPOOL space at the end of the report period.

### Warning message

The following warning message is issued by the SPLW sampler:

```
PWSSPLW1 iiiiiiii (jjjjjjjj) nnnnnnnn Waiting for SPOOL Space
```

Message PWSSPLW1 indicates that the specified address space is waiting for SPOOL space, where
The SRM sampler monitors the number of address spaces in use as a percentage of the MAXUSER setting in SYS1.PARMLIB member IEASYS00.

This sampler also monitors the system Unreferenced Interval Count (UIC) based on user-specified low and high thresholds.

**Syntax**

```
MAXUSER=nn - LOWUIC=nn - HIGHLUIC=nn
```

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length in intervals of the sample period; the default sample period is 30 seconds (1 interval)</td>
</tr>
<tr>
<td>MAXUSER=nn</td>
<td>is a percentage of the MAXUSER setting in SYS1.PARMLIB&lt;br&gt;A warning message is produced when the percentage of active address spaces equals or exceeds the specified value during the interval.</td>
</tr>
<tr>
<td>LOWUIC=nn</td>
<td>is the low UIC threshold</td>
</tr>
<tr>
<td>HIGHLUIC=nn</td>
<td>is the high UIC threshold</td>
</tr>
</tbody>
</table>

**Note**

If you omit a threshold specification or specify zero, no threshold monitoring is performed.

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at 30-second intervals if the number of users exceeds 80 percent of
the MAXUSER setting in SYS1.PARMLIB member IEASYS00, or the system Unreferenced Interval Count drops below 10 or exceeds 255:

SRM 1,MAXUSER=80,LOWUIC=10,HIGHUIC=255

**Warning messages**

The following warning messages are issued by the SRM sampler:

PWSSRM01 *WARNING* System MAXUSER count is mmmmmmmmm; current count is ccccccccc

Message PWSSRM01 indicates that the percentage of active address spaces equals or exceeds the MAXUSER percentage specified in the SRM control statement, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmm</td>
<td>is the MAXUSER setting in SYS1.PARMLIB</td>
</tr>
<tr>
<td>ccccccccc</td>
<td>is the number of active address spaces</td>
</tr>
</tbody>
</table>

PWSSRM02 *WARNING* System UIC sssssssss is below LOWUIC threshold of ttttttttt

Message PWSSRM02 indicates that the system UIC count is below the LOWUIC threshold specified in the SRM control statement, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sssssss</td>
<td>is the system's UIC count</td>
</tr>
<tr>
<td>ttttttt</td>
<td>is the LOWUIC threshold specified in the SRM control statement</td>
</tr>
</tbody>
</table>

PWSSRM03 *WARNING* System UIC sssssssss is above HIGHUIC threshold of ttttttttt

Message PWSSRM03 indicates that the system UIC count has exceeded the HIGHUIC threshold specified in the SRM control statement, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sssssss</td>
<td>is the system's UIC count</td>
</tr>
<tr>
<td>ttttttt</td>
<td>is the HIGHUIC threshold specified in the SRM control statement</td>
</tr>
</tbody>
</table>
The Swap Rate (SWP) sampler monitors system swap-out rate and produces a warning when the number of address spaces swapped out per minute exceeds a specified value.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( int )</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>( rate )</td>
<td>is the swap rate per minute measured over one reporting interval that triggers a warning message</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the swap-out rate exceeds 30 swaps-per-minute over the preceding two minutes:

```
SWP 4,30
```

**Warning message**

The following warning message is issued by the SWP sampler:

```
PWSSWP01 *WARNING* Swap-out rate is xxx swaps per minute
```

Message PWSSWP01 indicates that the swap rate has exceeded the warning threshold.
The TSO CPU (TCPU) sampler produces a warning whenever a TSO address space is detected that is using more than a specified percentage of the total CPU time during a reporting interval.

The message can be suppressed if a given threshold for total CPU usage is not reached during a reporting interval. This sampler also notifies the operator when TSO address spaces are using an excessive percentage of CPU time.

**Note**
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

![Syntax Diagram]

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system CPU time that an address space must be responsible for before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of address spaces to be listed whenever a CPU overload condition is detected; the default is 5</td>
</tr>
<tr>
<td>minpct</td>
<td>is the system CPU-busy percentage that must be reached before exception messages are issued; the default is zero</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays up to five warning messages at two-minute intervals for each interval in which the total time the CPU is busy increases 50 percent over the preceding interval:

```
TCPU 4,35,5,50
```

A warning is issued for any TSO address space that has been active 35 percent of the total CPU busy time.
**Warning message**

The following warning message is issued by the TCPU sampler:

PWSTCPU1 *WARNING* CPU USAGE IS ppp.p% FOR TSU jjjjjjjj

Message PWSTCPU1 indicates that the specified address space has exceeded the warning threshold for percentage of system CPU busy time, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppp.p</td>
<td>is the address space busy percentage</td>
</tr>
<tr>
<td>jjjjjjjj</td>
<td>is the address space name</td>
</tr>
</tbody>
</table>

**TIO**

The TSO I/O (TIO) sampler produces a warning whenever a specified TSO address space is detected that is using more than a specified percentage of the total system I/O activity in a reporting interval.

The message can be suppressed if a threshold for the total system I/O rate is not reached during a reporting interval. This sampler also notifies the operator when the I/O for a particular TSO address space exceeds the threshold.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system I/O for which a TSO address space must be responsible before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of TSO address spaces to be listed whenever a TSO I/O overload condition is detected; the default is 5</td>
</tr>
<tr>
<td>rate</td>
<td>is the rate of EXCPs per second that must be reached before exception messages are issued; the default is 25</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays up to eight warning messages (at two-minute intervals) for TSO address spaces that generated at least 5% of the I/O's during the interval, when the system I/O rate was at least 50 EXCPs-per-second:

```
TIO 4,5,8,50
```

A warning is issued for any TSO address space doing more than 25 percent of the I/O.

**Warning message**

The following warning message is issued by the TIO sampler:

```
PWSTIO01 *WARNING* I/O RATE FOR TSU nnnnnnnn IS xxx% OF SYSTEM TOTAL
```

Message PWSTIO01 indicates that the specified address space has exceeded the warning threshold for percentage of system I/O, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>is the address space name</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percent of the total system I/O rate</td>
</tr>
</tbody>
</table>

**TPAG**

The TSO Paging (TPAG) sampler produces a warning when a specified TSO address space is detected that is using more than a specified percentage of the total system paging activity during a reporting interval.

The warning can be suppressed if a given threshold for the total system paging rate is not reached during a reporting interval. This sampler also produces a warning when the paging for a particular TSO address space exceeds the threshold.
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system paging for which a TSO address space must be responsible before an exception message is issued for that address space</td>
</tr>
<tr>
<td>rptno</td>
<td>is the maximum number of TSO address spaces to be listed whenever a TSO-paging overload condition is detected. The default is 5.</td>
</tr>
<tr>
<td>rate</td>
<td>is the paging rate per second that must be reached before exception messages are issued; the default is a system paging rate greater than 25 EXCPs-per-second</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays up to five warning messages at two-minute intervals if the system-paging rate exceeds 50 pages-per-second over the preceding two minutes:

TPAG 4,25,5,50

A warning is issued for any TSO address space that is doing more than 25 percent of the paging.

Warning messages

The following warning messages are issued by the TPAG sampler:

PWSTPAG0 *WARNING* TOTAL SYSTEM PAGING RATE IS NOW ppp PAGES/SEC

Message PWSTPAG0 indicates that the system-paging rate has exceeded the warning threshold.

PWSTPAG1 *WARNING* PAGING RATE FOR TSU nnnnnnnn IS xxx% OF TOTAL
Message PWSTPAG1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnn</td>
<td>is the TSO user ID</td>
</tr>
<tr>
<td>xxx</td>
<td>is the percentage of the system total</td>
</tr>
</tbody>
</table>

**TSRV**

The TSO Service rate (TSRV) sampler produces a warning when a TSO address space uses more than the specified percentage of one or more SRM service measures during a reporting interval.

The warning can be suppressed if a given total system SRM rate threshold is not reached during a reporting interval.

You can specify multiple TSRV sampler statements. For example, you might want to have one TSRV statement for each measure.

**Note**

You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of total system SRM-component usage that TSO must be responsible for before an exception message is issued</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><em>measure</em></td>
<td>is the SRM service measure to be monitored:</td>
</tr>
<tr>
<td></td>
<td>C indicates CPU service units</td>
</tr>
<tr>
<td></td>
<td>I indicates I/O service units</td>
</tr>
<tr>
<td></td>
<td>M indicates MSO service units</td>
</tr>
<tr>
<td></td>
<td>T indicates total service units; the default</td>
</tr>
<tr>
<td><em>rptno</em></td>
<td>is the maximum number of address spaces to be listed whenever a TSO-service overload condition is detected; the default is 5</td>
</tr>
<tr>
<td><em>minrate</em></td>
<td>is the system SRM service-consumption rate that must be reached before exception messages are issued</td>
</tr>
<tr>
<td></td>
<td>The default is a system service rate greater than zero SUs-per-second.</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays up to five warning messages at two-minute intervals whenever an address space being monitored utilizes more than 35 percent of the total SRM components service during a reporting interval:

**TSRV 4,35,4,5,50**

This message can be suppressed if the system service rate threshold of 50 SUs-per-second is not reached during a reporting interval.

**Warning message**

The following warning message is issued by the TSRV sampler:

**PWSTSRV1 *WARNING* mmm SERVICE RATE FOR TSU jjjjjjjjj IS ppp% OF SYSTEM TOTAL**

Message PWSTSRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>mmm</em></td>
<td>is the SRM service component being monitored</td>
</tr>
<tr>
<td><em>jjjjjjjj</em></td>
<td>is the address space name</td>
</tr>
<tr>
<td><em>ppp</em></td>
<td>is the address space SRM component use as a percentage of system total</td>
</tr>
</tbody>
</table>
This Exception Monitor sampler can only be used from the MainView for z/OS product.

The Workload Delay sampler produces a warning when a workload’s delay meets or exceeds a specified threshold. A delay is the amount of time that a workload tried to use one or more resources and could not.

**Note**
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 23.

**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the number of intervals between report periods; the default is 30 seconds (one interval)</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the workload to be monitored</td>
</tr>
<tr>
<td>name*</td>
<td>indicates a partial workload name</td>
</tr>
<tr>
<td></td>
<td>You can specify partial names by using a wildcard character (an asterisk).</td>
</tr>
<tr>
<td>percent</td>
<td>represents the minimum percentage of delay</td>
</tr>
<tr>
<td></td>
<td>Any delay greater than this number generates a warning message.</td>
</tr>
<tr>
<td></td>
<td>Delays are calculated for the current data collection interval as defined by MainView for z/OS. That is, if a data collection interval is defined as 15 minutes, the workload delay percentage is calculated for the current 15-minute period, regardless of how many minutes have elapsed.</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages every three minutes if any workload starting with the letters MAC is delayed more than 30 percent of the current MainView for z/OS data collection interval:

WDLY 6,MAC*,30
Warning message

The following warning message is issued by the WDLY sampler:

PWSWDLY0 *WARNING* Workload wwwwwww Delayed xxx.xx%

Message PWSWDLY0 indicates that workload wwwwwww has been delayed xxx.xx percent of the current MainView for z/OS data collection interval.

WOBJ

The Workload Objective (WOBJ) sampler produces a warning when a workload is not meeting its specified percentage of service objectives.

This Exception Monitor sampler can be used only from the MainView for z/OS product.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the number of intervals between report periods; the default is 30 seconds (one interval)</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the workload to be monitored</td>
</tr>
<tr>
<td>name*</td>
<td>indicates a partial workload name; you can specify partial names by using a wildcard character (an asterisk)</td>
</tr>
<tr>
<td>percent</td>
<td>represents the threshold for workload service objectives, in decimal numbers If the workload’s percentage of satisfied service objectives is equal to or less than the number you specify, a warning message is issued. The workload’s service objectives are calculated for the current data collection interval as defined by MainView for z/OS.</td>
</tr>
</tbody>
</table>
Example

Assuming a report interval of 30 seconds, the following control statement displays warning messages every two minutes if any workload starting with the letters MAC meets 30 percent or less of its service objectives:

WOBJ 4,MAC*,30

Warning message

The following warning message is issued by the WOBJ sampler:

PWSWOBJ0 *WARNING* Workload wwwwwwwww at xxx% of its objective

Message PWSWOBJ0 indicates that workload wwwwwwwww has met xxx percent of its objective.

WRT

The Workload Response Time (WRT) sampler produces a warning when the response time for workloads that match your specified selection criteria reaches or exceeds the threshold you specified.

This Exception Monitor sampler can be used only from the MainView for z/OS product.

Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the report period in intervals; the default is 30 seconds (one interval)</td>
</tr>
<tr>
<td>name</td>
<td>is the name of the workload to be monitored</td>
</tr>
</tbody>
</table>
| name*     | indicates a partial workload name  
You can specify partial names by using a wildcard character (an asterisk), which indicates that only the characters prior to the asterisk are to be used for comparison. |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>period</td>
<td>represents the performance period for the workload, which is to be used when checking the response time against the threshold value.</td>
</tr>
<tr>
<td>seconds</td>
<td>is the response-time threshold expressed in hundredths of a second.</td>
</tr>
</tbody>
</table>

### Example

Assuming a report interval of 30 seconds, this example illustrates the use of four separate sampler definitions:

- **WRT 2,TSO*,1,70**
- **WRT 2,TSO*,2,200**
- **WRT 2,TSO*,3,300**
- **WRT 2,TSO*,4,400**

Each definition defines a response-time threshold for a specific performance period. A partial name of `tso*` is specified for each, indicating that any workload name that begins with `tso` is to be considered.

- The first definition indicates that a warning message should be issued if the response time for performance period 1 is 0.7 seconds or greater.
- The threshold for performance group 2 is two seconds.
- The threshold for performance group 3 is three seconds.
- The threshold for performance group 4 is four seconds.

### Warning message

The following warning message is issued by the WRT sampler:

**PWSWRT00  *WARNING* Workload wwwwwwwww response time is xxx.xx seconds**

Message PWSWRT00 indicates that the overall response time for workload `wwwwwwwww` is `xxx.xx` seconds.

### WTO

The WTO Buffer Usage (WTO) sampler monitors usage of console buffers and anticipates problems caused by a backlog of messages to be displayed on a system console.
**Syntax**

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td>percent</td>
<td>is the percentage of console buffers in use that causes a warning message to be issued; the default threshold is 75 percent</td>
</tr>
</tbody>
</table>

**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message every 30 seconds when the use of console buffers exceeds 80 percent:

```
WTO 1,80
```

**Warning message**

The following warning message is issued by the WTO sampler:

```
PWSWTO01 *WARNING* CONSOLE BUFFER USAGE IS xxx%
```

Message PWSWTO01 indicates that the specified threshold has been exceeded. You should check the status of the consoles.

**WTOR**

The WTOR sampler monitors usage of console reply buffers and displays a warning message when the user-specified threshold is reached.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>int</em></td>
<td>is the length of the sample period in intervals; the default sample period is 30 seconds (one interval)</td>
</tr>
<tr>
<td><em>percent</em></td>
<td>is the percentage of console buffers in use that causes a warning message to be issued; the default threshold is 75 percent</td>
</tr>
</tbody>
</table>

Example

Assuming a report interval of 30 seconds, the following control statement displays a warning message every 30 seconds when the use of console buffers exceeds 80 percent:

```plaintext
WTOR 1,80
```

Warning message

The following warning message is issued by the WTOR sampler:

```plaintext
PWSWTOR1 *WARNING* WTOR BUFFER USAGE IS xxx%
```

Message PWSWTOR1 indicates that the specified threshold has been exceeded. You should check the status of the consoles.

XCPU

The XCPU sampler monitors CPU utilization for the entire Plex and produces a warning when the total CPU utilization is outside of the limits that you specified.

Plex overhead is included in total CPU utilization.
Syntax

The parameters are defined as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>int</em></td>
<td><em>is the length of the sample period in intervals; the default is 30 seconds (one interval)</em></td>
</tr>
</tbody>
</table>
| *lowpct*   | *is the low threshold percent*  
A value of 0 (zero) indicates no low threshold checking will be performed  
A warning message is issued if the total Plex CPU utilization during the period is less than the specified percentage. Plex dispatch overhead is included in CPU utilization. |
| *highpct*  | *is the high threshold percent*  
A value of 0 (zero) indicates no high threshold checking will be performed  
A warning message is issued if the total Plex CPU utilization during the period equals or exceeds the specified percentage. Plex dispatch overhead is included in CPU utilization. |

**Example 1**

*XCPU,2,40*

The reporting period is two intervals (one minute if the default interval is used). A warning message will be issued if the Plex is less than 40% utilized during the period. No high threshold checking will be performed.

**Example 2**

*XCPU,10,0,90*

The reporting period is 10 intervals (five minutes if the default interval is used). A warning message will be issued if the Plex equals or exceeds 90% utilized during the period. No low threshold checking is to be performed.

**Example 3**

*XCPU,10,40,90*

The reporting period is 10 intervals (five minutes if the default interval is used). A warning message will be issued if the Plex is less than 40% or more than 90% utilized during the period.
Warning messages

The following warning messages are issued by the XCPU sampler:

PWSXCPU1 *WARNING* Total PLEX Utilization is nn.nn%

Message PWSXCPU1 is issued when the total Plex utilization is less than the lowpct value during the period.

PWSXCPU2 *WARNING* Total PLEX Utilization is nn.nn%

Message PWSXCPU2 is issued when the total Plex utilization is more than the highpct value during the period.
Index

@ symbol
  defined as special character 50
  for SYSPROG services commands 114
$$INEXCL
  syntax rules 23
$$INLGxx 145
$$INSYS0
  parameters 24
  syntax rules 24
$$INTKxx 264
$$RPJOBS
  syntax rules 29
$$RPTEXT
  syntax rules 31
$$XENQ
  syntax rules 32
$$XRES
  syntax rules 33

64-bit addresses, facilitating their use 52

A

accessing SYSPROG Services
  from an operator console 42
  from easy menus 38
  from EZMS PROG 38
  from SYSPROG Services Menu panel 38, 39
  from z/OS and USS Solutions panel 36
  with the MODIFY command 42
ACPU sampler 282
  excluding job names from exception reporting 23
ACTIVATE, LNKLST function 137
active task queue, monitoring 307
address expression, using symbolic name in place of 109

address space
  frames allocated to 217
  hung 124
  ID, displaying 267
  identifying malfunctions 169
  listing regions 190
  making nonswappable 91
  making swappable 174
  modules loaded in 70
  pending timer interrupts 262
  private storage allocated 203
  private, subpool allocations 190
  queues maintained in ASVT 64
  quiescing 111
  reusing 294
  slots available for starting new address spaces 325, 326
  summarizing virtual storage allocations 190
  SVC dump 222
  target 51
  terminating 116
  terminating tasks 85
  TSO. See TSO address spaces, monitoring 346
  wait limit 148
  waiting for SPOOL space 342
  warning message 342
  with above-the-bar memory objects 133
Address Space CPU (ACPU) sampler 282
Address Space I/O (AIO) sampler 283
Address Space I/O Rate (AIOR) sampler 285
Address Space Paging (APAG) sampler 287
Address Space Paging Rate (APGR) sampler 288
Address Space Service Rate Percentage (ASRV) sampler 290
Address Space Service Unit Rate (ASSR) sampler 292
address space token, displaying 247
address space, displaying
allocated devices [address space aa] 256
CPU time limits [address space aa] 135
device usage [address space aa] 270
frame allocation [address space aa] 217
Large Virtual Memory Objects [address space aa] 176
LSQA usage [address space aa] 190
request blocks (RB) [address space aa] 252
shared memory objects [address space aa] 176
status [address space aa] 246
step information [address space aa] 148, 195
storage allocations by subpool and key [address space aa] 190
task control blocks (TCB) [address space aa] 252
tasks [address space aa] 85
TQE [address space aa] 262

Advanced Early Warning (AEW) Exception
Monitor samplers. See Exception Monitor samplers 19
AEW sampler, see WARNING 273
AIO sampler 283
   excluding job names from exception reporting 23
AIOR sampler 285
   excluding job names from exception reporting 23
APAG sampler 287
   excluding job names from exception reporting 23
APF authorization 65
APF list
   adding entries 56
   changing the volser of existing entries 56
   updating 56
APF service 56
APGR sampler 288

excluding job names from exception reporting 23
ASID, nonreusable 64
ASM service 59
ASRV sampler 290
   excluding job names from exception reporting 23
ASSR sampler 292
   excluding job names from exception reporting 23
asterisk, used in an expression 50
ASVT sampler 294
ASVT service 64
AUTHCMD (CMD) authorization list 65
authorization lists
   AUTHCMD 65
   AUTHPGM 65
   AUTHTSF 65
   building 65
   modifying 65
   restoring 65
AUTHPGM (PGM) authorization list 65
AUTHTSF (TSF) authorization list 65
AUTHTSO service 65
AutoOPERATOR. See MainView AutoOPERATOR 20
auxiliary storage, managing 59
Available Queue, monitoring 294
available space on DASD volumes, monitoring 340

B

batch address spaces, monitoring
   CPU usage 295
   I/O usage 297, 304
   paging activity 298
   SRM component usage 300
Batch CPU (BCPU) sampler 295
Batch I/O (BIO) sampler 297
Batch Job mode 18
Batch Paging (BPAG) sampler 298
Batch Service (BSRV) sampler 300
BBPARM library members
executing commands listed in 114
PWSCPMMxx 33
PWSCPMMxx, setting report and sample intervals 319
BBSAMP library member, SYSPROJG 45
BBXS service, displaying status 68
BCPU sampler 295
excluding job names from exception reporting 23
BIO sampler 297, 304
excluding job names from exception reporting 23
block size, displaying for a specified data set 92
BMC Software, subsystem service, displaying status 68
BPAG sampler 298
excluding job names from exception reporting 23
BSRV sampler 300
excluding job names from exception reporting 23
buffers, console, monitoring 355

C

CADP sampler 302
CDE service 70
CHA sampler 303
Channel (CHA) sampler 303
channel paths, monitoring
    CADP 302
    CHA 303
    MIO 160
    MONITOR 164
    MPATH 165
character string, displaying location of occurrences 155
CLEAR service 72
CNSL service 74
comma
    separating service name and parameters 13
    used as placeholder 15
COMMAND service 75
commands
    invoking standard sequences 114
    syntax 12, 13
Common Area Dataspaces (CAD) Percentage sampler 300
Common Area Storage. See CSA storage 79
common storage
    frames allocated to 217
    freeing 117
    monitoring usage by job name 309
comparison string, specifying start and length 53
conflicts, enqueue 106, 208
console buffers, monitoring 355, 356
console reply buffers, number in use 76
console, nonoperational 76
CONSOLES service 76
Contents Directory Entry (CDE), listing for address spaces 70
control blocks
    ASVT 64
    LPA active queue 143
    SRM 242
    Subsystem Vector Table (SSVT) 244
conventions, documentation 12
CP commands, entering 272
CPU active task queue, monitoring 307
CPU busy percentage 201
CPU model number 123
CPU Queue Depth (CQD) sampler 307
CPU queue depth, monitoring 307
CPU sampler
    description 305
    excluding job names from exception reporting 23
CPU serial number 123
CPU service 77
CPU time limits 135
CPU usage
    displaying by job name 77
    monitoring address spaces 282
    monitoring batch address spaces 295
    monitoring by system 305
    monitoring for entire Plex 357
    monitoring host partition 330
    monitoring TSO address spaces 346
    warning message 282, 330, 357
CQD sampler 307
critical paging (CPAG) sampler 304
CSA and SQA Utilization (CSA) sampler 307
CSA sampler 307

Index 363
CSA service 79
CSA storage
  displaying usage 79
  monitoring usage 307
  monitoring usage by job name 309
CSMJ sampler 309
CSMON 248
CSNJ sampler
  excluding job names from exception reporting 23
CTCB service 85
  current date, displaying 255

D

DASD Free Space (SPA) sampler 340
DASD I/O Queue (DIOQ) sampler 311
DASD Reserves (RES) sampler 337
DASD volumes
  determining cause of poor performance 215
  displaying information about mount requests 169
  evaluating contention 163
  not ready 313
  under control of SMS 230
DASD volumes, displaying
  available space 237
  pending mount requests 169
  reserve activity 215
DASD volumes, monitoring
  available space 340
  I/O queue depth 311
  pending mount requests 324
  reserve activity 337
  using DNR sampler 313
DASD-mount requests, monitoring 324
data sets
  displaying information about 92
  usage by address space 256
data sets, system dumps
  clearing 72
  displaying title, date, and time 251
  SDUMP 222
date, displaying 255
DCPU service 201
DD names, listing members within a concatenation 256
DEACT, LNKLST function 137
default values, using commas as placeholders 15
delay, workload, monitoring 352
DEV sampler 320
DEVIATN service 87
Device Busy (DEV) sampler 310
Device Not Ready (DNR) sampler 313
devices
  allocated to an address space 256
  detecting potential performance delays 124
  identifying excessive contention 310
  monitoring 152, 164, 320
  type, displaying information about 92, 268
  used by address space 270
DFLIST service 90
diagnosing problems using the BBXS service 68
DIOQ sampler 311
Disabled Interrupt Exit (DIE) routines 262
DNR sampler 313
documentation information 11
domains, displaying processor usage 201
DONTSWAP service
  description 91
  reversing effects using OKSWAP 174
DSNAME service 92
DSSUM service 94
dump data sets
  clearing 72
  displaying title, date, and time 251
  SDUMP 222
DUMP service
  and the Last-Referenced Storage Pointer (LRSP) 50
  description 96
DVTN sampler 314

E
easy menus
  EZMSPROG 38
  EZMZOS 38
ECSA storage
  displaying usage 79
  monitoring usage by job name 309
EDTINFO service 100
elapsed time, monitoring jobs 320
electronic documentation 11
Eligible Device Table (EDT) 100
ENABLE command, loading exit routine into storage 226
END OF SERVICE message 24
ENQ sampler
description 316
excluding job names from exception reporting 23
excluding names from warning messages 32
Enqueue Conflicts (ENQ) sampler. See ENQ
sampler 316
enqueues
conflicts, monitoring 316
displaying conflicts 106
excluding names from display 32, 33
ENQUEUES service 106
tenqueues, displaying conflicts 208
entering commands 13
EQUATE service
description 109
symbol names for memory objects 51
ESCLASS service 111
esoteric unit names 100
ESQA storage, monitoring usage by jobname 309
Exception Monitor
controlling 273
excluding job names from 23
generating AO Alerts 20
messages, controlling 24
setting the report interval 319
setting the sample interval 319
storing as members in BBPARM 33
Exception Monitor samplers
ACPU 282
AIO 283
AIOR 285
APAG 287
APGR 288
ASRV 290
ASSR 292
ASVT 294
BCPU 295
BIO 297, 304
BPAG 298
BSRV 300
CADP 302
CHA 303
CPU 305
CQD 307
CSA 307
CSMJ 309
definition 19
DEV 320
DIOQ 311
DNR 313
DVTN 314
ENQ 316
FIX 318
INT 319
JET 320
JSU 321
JVR 323
MTP 324
NRQ 325
NRQP 326
OUT 328
PAG 329
PCPU 330
PER 332
PGQ 333, 334
REP 336
RES 337
RSU 338
SMF 339
SPA 340
SPLW 342
SRM 343
SWP 345
TCPU 346
TIO 347
TPAG 348
TSRV 350
WDLY 352
EXCPs, monitoring address spaces 285
EXECUTE service 114
executing commands 114
exempting from reply scan 29
exit point 226
exit routines 226
EXIT service 116
expressions 48
Extended Common Service Area. See ECSA storage 79
Extended SVC Router (ESR) table 143
extended system link-pack area. See LPA (link-pack area) 143
Extended System Queue Area. See ESQA storage 79
extents, displaying number for a specified data set 92

F
FCOMMON service 117
FINDMBR service 118
FIX sampler 318
Fixed Pages (FIX) sampler 318
fixed pages, monitoring 318
frame allocation
  displaying for an address space 217
  monitoring usage by job 321
free space
  fragmented 190
  on DASD volumes, monitoring 340

G
G scale suffix 51
generic unit names 100
Gi scale suffix 51
Gigabytes scale suffix
  G 51
  Gi 51
Greenwich Mean Time (GMT), displaying 255
Gregorian time format, displaying 255

H
Help
  online 11
HELP service 120
hexadecimal format, displaying storage contents 96
hexadecimal string, displaying location of occurrences 155

I
I/O bottlenecks, detecting 160
I/O interrupt activity, displaying distribution among processors 183
I/O queue depth
  monitoring DASD devices 311
  monitoring page data sets 333, 334
I/O service 124
I/O system, monitoring
  performance 160
  warning message 283
I/O usage
  displaying 124
  monitoring address spaces 152, 283
  monitoring batch address spaces 297, 304
  monitoring devices 152
  monitoring TSO address spaces 347
I/O, outstanding 260
IFA service 122
inactive jobs that should be active 87, 314
indirection indicators 49
INFO service 123
Initialization Parameter Area (IPA) 127
INT service 319
Interval (INT) sampler 319
IO service 124
IOS driver codes 124, 260
IPL, displaying information about 123
IPLDATA service 127
ISMF service 129
ISPF EDIT 106

J
JCL, submitting for operations jobs 248
JES commands, invoking standard sequences 114
JET sampler
description 320
excluding job names from exception reporting 23
JITOKEN service 131
Job Awaiting V=R Region (JVR) sampler 323
Job Elapsed Time (JET) sampler 320
job entry subsystem, submitting jobs to 248
Job Storage Usage Real & Auxiliary (JSU) sampler 321
jobs
activity, monitoring 77
awaiting V=R region 323
displaying limits 135
elapsed time 320
frame usage 321
inactive, should be active 87, 314
monitoring elapsed time 320
monopolizing devices and channel paths 160
overuse of real and auxiliary storage 321
resuming after quiescing 111
slot usage 321
submitting JCL 248
swapped out of memory 328
use of VIO slots 271
with outstanding WTORs 29
Jobs Swapped Out (OUT) sampler 328
JSTORM service 133
JSU sampler
description 321
excluding job names from exception reporting 23
Julian time format, displaying 255
JVR sampler
description 323
excluding job names from exception reporting 23

K
K scale suffix 51
Ki scale suffix 51
Kilobytes scale suffix
  K 51
  Ki 51

L
LABEL service 134
Large Virtual Memory Objects 176
Last-Referenced Storage Pointer (LRSP) 50
LCPU service 135
line number, displaying 267
list of services 38, 39
LIST, LNKLIST function 137
LNKLST service 137
LNKLST sets 137
LOADLPA service 143
LOCATE, LNKLIST function 137
LOGGING service 145
logging system 145
logical control units, monitoring 163
logical partitions (LPARs), displaying processor usage 201
logical record length, displaying for a specified data set 92
LOGREC data sets, reading software records 233
LOGSTREAM, reading software records from 233
LPA (link-pack area)
  active queue 143
LRSP (Last-Referenced Storage Pointer) 50
LWAIT service 148
LX service 150

M
M scale suffix 51
main memory, modifying 277
main storage 277
MainView AutoOPERATOR monitors and responds to warning messages 20
SYSPROG Services available from 55
major name (Q name) 106
managing auxiliary storage, using ASM 59
master console, issuing commands from a TSO terminal 152
MCOMMAND service 152
MDEVICE service 152
Megabytes scale suffix
  M 51
  Mi 51
memory locations
  displaying 96
  searching for strings 155
memory object
  associated with an address space 176
  searching for strings 172
MEMSCAN service 155
messages
backlog, displayed on system console 355
how displayed 24
identification numbers 24
to operator, outstanding 336
undisplayed, per system console 76
warning about exceeded threshold 281
warning, suppressing. See names of individual
samplers, suppress warning message 283
Mi scale suffix 51
minor name (R name) 106
MIO service 160
MLCU service 163
modes, operation 18
modules
adding in LPA 143
deleting 143
displaying LOGREC data 233
loaded in an address space 70
MONITOR service 164
monitoring
enqueue conflicts 316
thresholds 281
mount requests 169, 324
Mounts Pending (MTP) sampler 324
MPATH service 165
MSTORAGE service 167
MTP sampler 324
MTP service 169
Multiple Domain Facility (MDF) complex 201, 208
MVS commands, invoking standard sequences 114

N
Name/Token pair information 169
Non Reusable Queue (NRQ) sampler 325
Non Reusable Queue Percentage (NRQP) sampler 326
Non-reusable Replacement Queue, monitoring 294
non-TP I/O, displaying 124
non-VIO slots 59
nonswappable address space 91
NRQ sampler 325
NRQP sampler 326
NTP service 169

O
OFIND service 172
OKSWAP service 174
OLIST service
creating symbolic names before using OFIND 172
description 176
symbol names for memory objects 51
online Help 11
operating system, displaying information about 123
operation
modes 18
parameters 180
operator commands, simulating from TSO terminal 75
operator console, simulating 74
operator messages, outstanding 336
option settings 225
OUT sampler
description 328
excluding job names from exception reporting 23
outstanding I/O 260
outstanding replies
displaying 214, 336
monitoring 336

P
P scale suffix 51
PAG sampler 329
Page Data Set I/O Queue (PGQ) sampler 333
page data sets
monitoring I/O queue depth 333, 334
using ASM service 59
page fixing, monitoring 318
page reclaim rate 179
paging activity, analyzing 179
paging activity, monitoring
address spaces 287, 288
batch address spaces 298
rate 329
TSO address spaces 348
warning message 287
Paging Rate (PAG) sampler 329
PAGING service 179
parameters
defining run-time 24
in syntax diagrams 13
operation 180
system initialization 180
PARMLIST
service 180
PAS operating mode 18
path definition 268
path failures, detecting 165
PCPU sampler 330
PCPU service 201
pending mount requests
displaying 169
monitoring 324
PER sampler 332
performance monitoring of I/O subsystem 160
Petabytes scale suffix
  P 51
  Pi 51
PF key, displaying or changing definition of 182
PFKEY service 182
PGQ sampler 333, 334
Pi scale suffix 51
PIO service 183
placeholder, comma used as 15
POST service 185
PPT service 186
predefined symbols 51
PRIVATE service 190
private storage
  allocation of 190
  amount allocated 190, 203, 217
displaying 167
private storage (PVT) sampler 334
problem resolution
  from a TSO terminal 75
  simulating an operator console 74
  using the BBXS service 68
PROCESS, LNKLST function 137
processor usage by partition 201
program function key, displaying or changing
definition of 182
program properties table, displaying and updating 186
PROGRESS service 195
PRSM service 201, 208
PSTORAGE service 203
publications, related 11
PWSCPMxx
  suffix definition 24
  syntax rules 33
Q
Q name. See major name 106
queues
  ASVT control block 64
  ASVT sampler 294
  monitoring 294
R
R name. See minor name 106
RBs, displaying internal values 252
real storage
  displaying allocation 217
  monitoring status 338
Real Storage Usage (RSU) sampler 338
record format, displaying for a specified data set 92
record length, logical, displaying for a specified
data set 92
Recovery/Termination Manager 116, 176, 203
related publications 11
relative share 77
REP sampler 336
  excluding job names from exception reporting 23
replacement address 49
Replies Outstanding (REP) sampler 336
REPLEIS service 214
replies, monitoring 336
reply buffers
  console, monitoring 356
  number in use 76
reporting base interval, changing 319
request blocks, displaying internal values 252
RES sampler 337
  excluding job names from exception reporting 23
reserve activity
  displaying for DASD devices 215
  monitoring DASD devices 337
RESERVES service 215
resource conflict information, displaying 106
resource ownership information, displaying 106
response time, workload, monitoring 354
RSM service 217
RSU sampler 338
  excluding job names from exception reporting 23
Index 369
run-time parameters, defining 24
RWAIT parameter
  $INSYS0 24
  SETOP option 225

S

scale suffixes
  G 51
  Gi 51
  K 51
  Ki 51
  M 51
  Mi 51
  P 51
  Pi 51
  scaling decimal numbers 51
  T 51
  Ti 51

scanning memory and storage areas 155
SDUMF service 222
search string, specifying 53
service classes, changing, displaying, listing 111
service objectives, workload, monitoring 353
session information, displaying 267
session, displaying type used 267
Set Counter (SCT). See SCT service 221
SETOP service 225
settings, option 225
SLIP trap, PER sampler 332
slots
  monitoring number 294
  monitoring usage by job 321
SMF data sets
  displaying information about 129
  full 339
SMF exits
  deleting 226
  enabling 226
  routines 226
SMF recording, not active 339
SMF sampler 339
SMFEXIT service 226
SMS (system-managed storage), displaying DASD information 230
SMSLIST service 230
SOFTFRR service 233
software errors, displaying 233
software LOGRECs, summarizing 233
software problems, diagnosing using the BBXS service 68
SPA sampler 340
SPACE service 237
SPLW sampler 342
SPOOL space, warning message 342
SQA storage
  displaying usage 79
  frames allocated to 217
  monitoring usage 307, 309
SRB mode 96
SRM component usage
  monitoring address spaces 290, 292
  monitoring batch address spaces 300
  monitoring TSO address spaces 350
SRM sampler 343
SRM service
  components, monitoring 290, 292
  description 242
SSCVT (Subsystem Communications Vector Table) 244
SSVT (Subsystem Vector Table) 244
SSVT service 244
Start/SASI Queue, monitoring 294
Started Task mode 18, 45
Status of SMF Recording (SMF) sampler 339
STATUS service 246
STOKEN service 247
storage
  displaying contents in hexadecimal and character formats 96
  group names 230
  main 277
  private. See private storage 190
storage areas
  scanning for character or hexadecimal strings 155
  subject address 276
storage-constrained systems, tuning 190
string occurrences, creating symbol for 155
subject address storage area 276
subpool allocations within a private address space 190
Subsystem Communications Vector Table (SSCVT) 244
Subsystem Vector Table (SSVT) 244
subsystem, adding without having to re-IPL 244
suffix characters, used to scale decimal numbers 51
suppress warning message
batch address space paging too much 298
batch address space using too much CPU 295
batch address space using too much system I/O 297
batch address space using too much system SRM 300
elapsed time of jobs 320
enqueue conflicts for specific address spaces 316
excessive SRM service component resources used 290
excessive SRM service components used 292
I/O activity 283
jobs awaiting V=R region 323
jobs swapped out of memory 328
outstanding operator messages 336
overuse of real and auxiliary storage 321
TSO address space 346-348, 350
SVC dump
clearing 72
displaying date and title 251
taking 222
SVC table
planning installation 249
scanning 249
updating 143
SVCFIND service 249
Swap Rate (SWP) sampler 345
swap-out rate
monitoring 345
OUT sampler 328
SWP sampler 345
symbols
attributes 51
creating for string occurrences 155
defining, redefining, and displaying using EQUATE 109
overview 50
used in place of address expressions 109
used with MEMSCAN service 155
user-created 96
syntax diagrams, how to read 12, 13
syntax shortcuts to facilitate entering 64-bit addresses 52
syntax statement conventions 15
SYS1.PARMLIB members
CONFIGxx 87, 314
IEASYS00 343
IKJTSOxx 65
listing contents 180
TSO command 45
TSULIST 267
UCB 268
USING 270
VIO 271
VMCMD 272
WARNING 273
ZIIP 279
SYSPROG Services Menu panel 38, 39
SYSPROG services
  STOKEN 247
system activity, monitoring 77
system and non-system linkage indexes 150
system console, number of undisplayed messages 76
system contention caused by long-term reserves 215
system dump data sets
  clearing 72
  displaying date and title 251
  SDUMP 222
system initialization parameters 180
system paging rate, monitoring 329
system problems, solving 75
System Queue Area. See SQA storage 79
system resource information, displaying 242
system swap-out rate, monitoring 345
system-managed storage (SMS), displaying DASD information 230

T
T scale suffix 51
tape devices, displaying information about mount requests 169
tape labels, displaying contents 134
tape-mount requests, monitoring 324
target address space 51
task control block. See TCBs 252
TCB service 252
TCBs
  displaying internal values 252
  displaying storage allocations by subpool and key 190
TCPU 23
TCPU sampler 346
teleprocessing devices, displaying I/O 260
Terabytes scale suffix
T 51
Ti 51
Terminal I/O Task service 256
terminal name, displaying 267
terminating
  address space 116
  tasks in an address space 85
thresholds
  monitoring 281
Ti scale suffix 51
TIME service 255
time, displaying 255
Timer Queue Elements (TQE) 262
TIO sampler 347
  excluding job names from exception reporting 23
TIOT service 256
TOD service 258
TP lines, enabled 260
TPAG sampler 348
  excluding job names from exception reporting 23
TPIO service 260
TQE service 262
TRACK service 264
tracks, displaying for a specified data set 92
TSO
  Line mode 46
  operating mode 18
  See also authorization list[TSO:aaaa] 65
  session information 267
TSO address spaces, monitoring
  CPU usage 346
  I/O usage 347
  paging activity 348
  SRM component usage 350
TSO CPU (TCPU) sampler 346
TSO I/O (TIO) sampler 347
TSO Paging (TPAG) sampler 348
TSO Service (TSRV) sampler 350
TSRV sampler 350
  excluding job names from exception reporting 23
TSULIST service 267

U
UCB service 268
UNDEFINE, LNKLST function 137
underscore, used in 64-bit addresses 52
undisplayed messages per console services
   CONSOLES 76
unit control blocks (UCB), displaying 268
unit names, displaying information about 100
Unreferenced Interval Count (UIC) 343
UPDATE, LNKLST function 137
user ID
   displaying 267
   use of VIO slots 271
USERS, LNKLST function 137
USING service 270

V

V=R region, monitoring waiting jobs 323
VIO service 271
VIO slots
   ASM service 59
   displaying usage by job name or user ID 271
virtual console interface 272
virtual storage areas
   allocating for private storage 190
   displaying usage 167
   mapping 167
VM user, issuing CP commands through virtual
   console interface 272
VMCMD service 272
volser
   changing 56
   displaying for a specified data set 92
volume serial number
   changing 56
   displaying for a specified data set 92
volume serial status 268
volumes
   DASD, free space (SPA) sampler 340
   VSAM data sets 92

W

wait limit for an address space 148
wait state 76, 148
warning message

address space issuing EXCPs at high rate 285
address space using too much CPU time 282
address space using too much system I/O 283
address spaces in use 343
available address space vector table slots
   (ASIDs) 325, 326
available SPOOL space 342
average queue depth exceeding threshold 307
batch address space paging too much 298
batch address space using too much CPU 295
batch address space using too much system I/O 297
batch address space using too much system SRM 300
channel path exceeded threshold 303
console buffer usage 355
console reply buffer usage 356
CPU utilization for host partition 330
CSA or SQA usage exceeding threshold 307
DASD devices online but not ready 313
depth of I/Os queued to page data sets 333
elapsed time of jobs 320
enqueue conflicts 316
excessive device contention 310
excessive number of I/Os queued to DASD device 311
excessive paging during reporting interval 288
excessive rate of SRM service components 292
excessive SRM service component resources
   used 290
fixed pages 318
free space on DASD volumes 340
jobs awaiting V=R region 323
jobs meeting or exceeding thresholds 309
jobs swapped out of memory 328
outstanding operator messages 336
overuse of real and auxiliary storage 321
page fixing 318
paging activity 287
real storage use excessive 338
reserved status of devices in shared DASD system 337
slot number low 294
slots available for starting new address spaces 325, 326
SMF data, potential loss 339
suppressing. See suppress warning message 316
system paging rate 329
system swap-out rate 345
tape and DASD-mount requests 324
too much paging activity 287
TSO address space 346–348, 350
workload delay 352
workload response time 354
workload service objectives 353
WARNING service 273
WDLY sampler
  excluding job names from exception reporting 23
WDLY service 352
WHERE service 276
WOBJ sampler 353
workload
  monitoring delays 160, 352
  monitoring response time 354
  monitoring service objectives 353
Workload Delay sampler 352
Workload Objective (WOBJ) sampler 353
Workload Response Time (WRT) sampler 354
WRT sampler 354
WTO Buffer Usage (WTO) sampler 355
WTO sampler
  description 355
WTOR messages
  containing specific text 31
  displaying 214
WTOR sampler
  description 356
WTORs, exempting from reply scan 31

X

XCPU sampler 357

Z

z/OS and USS Solutions panel, accessing
  SYSPROG Services from 36
z/OS easy menu 38
zAAP processors 122, 279
ZAP service
  and the Last-Referenced Storage Pointer (LRSP) 50
  description 277
ZIIP service 279
zSeries Application Assist Processor (zAAP). See
  zAAP processors 122
zSeries Integrated Information Processor (zIIP)
  usage 279