MainView SYSPROG
Services User Guide and Reference

Supporting

Version 3.7 of MainView SYSPROG Services

September 2012
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 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA</td>
<td>1 713 918 8800 or 1 800 841 2031</td>
<td>1 713 918 8000</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>


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.

DB2, IBM, MVS, OS/390, VTAM, z/Architecture, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at 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 issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- 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. (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 issue

- 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, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
## Contents

**About this book**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related publications</td>
<td>15</td>
</tr>
<tr>
<td>Online Help</td>
<td>16</td>
</tr>
<tr>
<td>Conventions</td>
<td>16</td>
</tr>
<tr>
<td>Identifying address spaces</td>
<td>17</td>
</tr>
<tr>
<td>Expressions</td>
<td>18</td>
</tr>
<tr>
<td>Indirection indicators</td>
<td>19</td>
</tr>
<tr>
<td>Symbols</td>
<td>19</td>
</tr>
<tr>
<td>Last-Referenced Storage Pointer</td>
<td>20</td>
</tr>
<tr>
<td>Target address space</td>
<td>20</td>
</tr>
<tr>
<td>Scale suffixes</td>
<td>21</td>
</tr>
<tr>
<td>Working with 64-bit addresses</td>
<td>21</td>
</tr>
<tr>
<td>Specifying the start and length of a comparison string</td>
<td>22</td>
</tr>
<tr>
<td>Syntax diagrams</td>
<td>22</td>
</tr>
<tr>
<td>Understanding MainView SYSPROG Services command syntax</td>
<td>23</td>
</tr>
<tr>
<td>Using commas as placeholders for default values</td>
<td>24</td>
</tr>
</tbody>
</table>

**Chapter 1 Introducing MainView SYSPROG Services**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView SYSPROG Services features</td>
<td>25</td>
</tr>
<tr>
<td>Supported environments</td>
<td>26</td>
</tr>
<tr>
<td>Services</td>
<td>27</td>
</tr>
<tr>
<td>Exception Monitor samplers</td>
<td>28</td>
</tr>
<tr>
<td>How MainView SYSPROG Services works with other products</td>
<td>28</td>
</tr>
</tbody>
</table>

**Chapter 2 Controlling the MainView SYSPROG Services environment**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How MainView SYSPROG Services locates the parameter library members</td>
<td>29</td>
</tr>
<tr>
<td>PARMLIST: parameter member processing report</td>
<td>30</td>
</tr>
<tr>
<td>Use of symbols within members</td>
<td>30</td>
</tr>
<tr>
<td>Parameter library members</td>
<td>30</td>
</tr>
<tr>
<td>$$INEXCL</td>
<td>31</td>
</tr>
<tr>
<td>$$INNWTO</td>
<td>33</td>
</tr>
<tr>
<td>$$INSYS0</td>
<td>34</td>
</tr>
<tr>
<td>$$RPJOBS</td>
<td>37</td>
</tr>
<tr>
<td>$$RPTEXT</td>
<td>38</td>
</tr>
<tr>
<td>$$XENQ</td>
<td>39</td>
</tr>
<tr>
<td>$$XRES</td>
<td>40</td>
</tr>
<tr>
<td>PWSCPMxx</td>
<td>40</td>
</tr>
</tbody>
</table>
# Chapter 3 Using MainView SYSPROG Services

Accessing MainView SYSPROG Services ................................. 45  
Accessing MainView SYSPROG from the z/OS and USS Solutions Panel ................................. 45  
Accessing MainView SYSPROG Services from the MainView for z/OS easy menu ................................. 47  
Accessing Services from the SYSPROG Services Menu panel ................................. 49  
Executing MainView SYSPROG Services from an operator console ................................. 53  
Using the MODIFY (F) command ................................. 53  
Using a CMDID character ................................. 53  
Independent operations ................................. 54  
Executing MainView SYSPROG Services as a Started Task ................................. 54  
MainView SYSPROG Services as a TSO command ................................. 56  
MainView SYSPROG Services as a batch job ................................. 57  

# Chapter 4 Exception Monitor samplers

Introduction ................................. 61  
ACPU ................................. 62  
AIO ................................. 64  
AIOR ................................. 66  
APAG ................................. 67  
APGR ................................. 69  
ASRV ................................. 71  
ASSR ................................. 73  
ASVT ................................. 75  
BCPU ................................. 77  
BIO ................................. 78  
BPAG ................................. 80  
BSRV ................................. 82  
CADP ................................. 84  
CHA ................................. 85  
CPU ................................. 87  
CQD ................................. 89  
CSA ................................. 90  
CSMJ ................................. 93  
DEV ................................. 95  
DIOQ ................................. 97  
DNR ................................. 99  
DVTN ................................. 100  
ENQ ................................. 102  
FIX ................................. 103  
INT ................................. 105  
JET ................................. 106  
JSU ................................. 108  
JVR ................................. 110  
MTP ................................. 111  
NRQ ................................. 112  
NRQP ................................. 113  
OUT ................................. 115
Chapter 5 Services 149

AEW .................................................. 151
ANALYZE ........................................... 152
APF .................................................... 153
ASM .................................................... 157
ASVT ................................................... 164
AUTHITSO ........................................... 165
BBXS .................................................. 169
CDE .................................................... 171
CLEAR .................................................. 174
CNSL .................................................. 176
COMMAND ........................................ 178
CONSOLES ....................................... 179
CPU .................................................... 180
CSA .................................................... 182
CSMON ............................................... 185
CTCB .................................................. 185
DEVIATN ........................................... 188
DFLIST ............................................. 190
DONTSWAP ....................................... 192
DSNAME ........................................... 193
DSSUM ............................................... 195
DUMP .................................................. 198
DVIEW ............................................... 203
EDTINFO .......................................... 209
ENCLAVE ......................................... 215
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENQUEUEES</td>
<td>217</td>
</tr>
<tr>
<td>EQUATE</td>
<td>221</td>
</tr>
<tr>
<td>ESCLASS</td>
<td>223</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>228</td>
</tr>
<tr>
<td>EXIT</td>
<td>231</td>
</tr>
<tr>
<td>FCOMMON</td>
<td>232</td>
</tr>
<tr>
<td>FINDMBR</td>
<td>233</td>
</tr>
<tr>
<td>HELP</td>
<td>236</td>
</tr>
<tr>
<td>IFA</td>
<td>239</td>
</tr>
<tr>
<td>INFO</td>
<td>240</td>
</tr>
<tr>
<td>IO</td>
<td>242</td>
</tr>
<tr>
<td>IPLDATA</td>
<td>245</td>
</tr>
<tr>
<td>ISMF</td>
<td>249</td>
</tr>
<tr>
<td>JCPU</td>
<td>251</td>
</tr>
<tr>
<td>JITOKEN</td>
<td>253</td>
</tr>
<tr>
<td>JSTORM</td>
<td>254</td>
</tr>
<tr>
<td>LABEL</td>
<td>256</td>
</tr>
<tr>
<td>LCPU</td>
<td>257</td>
</tr>
<tr>
<td>LNKLST</td>
<td>259</td>
</tr>
<tr>
<td>LOADLPA</td>
<td>266</td>
</tr>
<tr>
<td>LOGGING</td>
<td>269</td>
</tr>
<tr>
<td>LWAIT</td>
<td>272</td>
</tr>
<tr>
<td>LX</td>
<td>275</td>
</tr>
<tr>
<td>MCOMMAND</td>
<td>279</td>
</tr>
<tr>
<td>MDEVICE</td>
<td>280</td>
</tr>
<tr>
<td>MEMSCAN</td>
<td>284</td>
</tr>
<tr>
<td>MIO</td>
<td>290</td>
</tr>
<tr>
<td>MLCU</td>
<td>293</td>
</tr>
<tr>
<td>MONITOR</td>
<td>295</td>
</tr>
<tr>
<td>MPATH</td>
<td>297</td>
</tr>
<tr>
<td>MSTORAGE</td>
<td>299</td>
</tr>
<tr>
<td>MTP</td>
<td>301</td>
</tr>
<tr>
<td>NTP</td>
<td>302</td>
</tr>
<tr>
<td>OFIND</td>
<td>306</td>
</tr>
<tr>
<td>OKSWAP</td>
<td>310</td>
</tr>
<tr>
<td>OLIST</td>
<td>312</td>
</tr>
<tr>
<td>PAGING</td>
<td>317</td>
</tr>
<tr>
<td>PARMLIST</td>
<td>318</td>
</tr>
<tr>
<td>PBUILD</td>
<td>320</td>
</tr>
<tr>
<td>PFKEY</td>
<td>321</td>
</tr>
<tr>
<td>PIO</td>
<td>323</td>
</tr>
<tr>
<td>PLIST</td>
<td>324</td>
</tr>
<tr>
<td>POST</td>
<td>325</td>
</tr>
<tr>
<td>PPT</td>
<td>326</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>331</td>
</tr>
<tr>
<td>PROGRESS</td>
<td>337</td>
</tr>
<tr>
<td>PRSM</td>
<td>342</td>
</tr>
<tr>
<td>PSTORAGE</td>
<td>346</td>
</tr>
<tr>
<td>QUEUE</td>
<td>352</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
</tr>
<tr>
<td>REPLIES</td>
<td>360</td>
</tr>
<tr>
<td>RESERVES</td>
<td>362</td>
</tr>
<tr>
<td>RSM</td>
<td>364</td>
</tr>
<tr>
<td>SCT</td>
<td>369</td>
</tr>
<tr>
<td>SDUMP</td>
<td>370</td>
</tr>
<tr>
<td>SETOP</td>
<td>374</td>
</tr>
<tr>
<td>SMFEXIT</td>
<td>376</td>
</tr>
<tr>
<td>SMSLIST</td>
<td>381</td>
</tr>
<tr>
<td>SOFTFRR</td>
<td>385</td>
</tr>
<tr>
<td>SP2</td>
<td>390</td>
</tr>
<tr>
<td>SPACE</td>
<td>393</td>
</tr>
<tr>
<td>SRM</td>
<td>398</td>
</tr>
<tr>
<td>SSVT</td>
<td>401</td>
</tr>
<tr>
<td>STATUS</td>
<td>403</td>
</tr>
<tr>
<td>SUBMIT</td>
<td>405</td>
</tr>
<tr>
<td>SVCFIND</td>
<td>407</td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>410</td>
</tr>
<tr>
<td>TCB</td>
<td>411</td>
</tr>
<tr>
<td>TIME</td>
<td>415</td>
</tr>
<tr>
<td>TIOI</td>
<td>417</td>
</tr>
<tr>
<td>TOD</td>
<td>420</td>
</tr>
<tr>
<td>TPIO</td>
<td>422</td>
</tr>
<tr>
<td>TQF</td>
<td>425</td>
</tr>
<tr>
<td>TRACK</td>
<td>427</td>
</tr>
<tr>
<td>TSULIST</td>
<td>431</td>
</tr>
<tr>
<td>UCB</td>
<td>433</td>
</tr>
<tr>
<td>USING</td>
<td>435</td>
</tr>
<tr>
<td>VIO</td>
<td>437</td>
</tr>
<tr>
<td>VMCMD</td>
<td>438</td>
</tr>
<tr>
<td>WARNING</td>
<td>439</td>
</tr>
<tr>
<td>WHERE</td>
<td>442</td>
</tr>
<tr>
<td>ZAP</td>
<td>444</td>
</tr>
<tr>
<td>ZIIP</td>
<td>446</td>
</tr>
</tbody>
</table>

**Appendix A**  Customizing MainView SYSPROG Services  447

- AutoCustomization  448
  - Invoking AutoCustomization  448
- Manual customization  449
  - Obtaining authorization for SYSPROG Services  449

**Appendix B**  Previous version of the Progress service  451

**Index**  455
Figures

MainView Selection Menu panel ................................. 45
z/OS and USS Solutions panel .................................. 46
SYSPROG Easy Menu (EZMSPS) .................................. 46
Select MainView SYSPROG Services from the Utilities category .................. 47
SYSPROG Fast Menu (EZMFPROG) ................................. 48
SYSPROG Services Menu panel .................................. 49
Service Information panel for DVIEW ............................ 51
SYSPROG Services Output panel for DVIEW ....................... 52
Sample JCL for MainView SYSPROG Services ..................... 54
Sample batch job for MainView SYSPROG Services .................. 57
Tables

MainView SYSPROG Services operation modes ........................................... 27
BBPARM library members ................................................................. 30
Keywords for member $$INSYS0 ......................................................... 34
Available MainView SYSPROG Services commands ............................. 50
Types of storage monitored by the FIX sampler .................................. 103
SPACE service keywords ................................................................. 395
TCB address symbols ....................................................................... 415
TQE flags ......................................................................................... 426
How TSF users obtain authorization to MainView SYSPROG Services ........ 450
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 Customer Support website at [http://www.bmc.com/support](http://www.bmc.com/support). Most product shipments also include the books on a documentation CD.

**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](http://www.adobe.com).

Related publications

From the BMC Support Central website ([http://www.bmc.com/support](http://www.bmc.com/support)), you can use either of the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center ([https://webapps.bmc.com/infocenter/index.jsp](https://webapps.bmc.com/infocenter/index.jsp)) to browse documentation sets, or to view video demos (short overviews of selected product concepts, tasks, or features)

- 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.
Online Help

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

Conventions

This book uses the following general conventions:

<table>
<thead>
<tr>
<th>Item</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>information that you are instructed to type</td>
<td>bold</td>
<td>Type <strong>EZMSPROG</strong> on the <strong>COMMAND</strong> line.</td>
</tr>
<tr>
<td>specific (standard) keyboard key names</td>
<td>bold</td>
<td>Press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>field names, option names, directories, file names</td>
<td>bold</td>
<td>If you specify <strong>SYSID=SMF</strong>, the SMF identifier in field <strong>SMCASID</strong> is used.</td>
</tr>
<tr>
<td>Web addresses, e-mail addresses</td>
<td>underlined blue text</td>
<td>The BMC Software home page is at <strong><a href="http://www.bmc.com">www.bmc.com</a></strong>.</td>
</tr>
<tr>
<td>view names, command names, nonspecific key names, keywords</td>
<td>uppercase</td>
<td>Use the <strong>HELP</strong> function key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use the <strong>WARN</strong> view when SYSPROG is operating within the PAS.</td>
</tr>
<tr>
<td>commands that can be shortened</td>
<td>required letters capitalized</td>
<td>To clear the screen, type <strong>RESet</strong>.</td>
</tr>
<tr>
<td>code examples, syntax statements, system messages, screen text</td>
<td>code typeface</td>
<td><strong>AMT000A ENTER SYSPROG Services COMMAND</strong></td>
</tr>
<tr>
<td>emphasized words, new terms, variables</td>
<td>italics</td>
<td>You must run the job on <strong>each</strong> system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keywords must be entered in the format <strong>keyword=value</strong>.</td>
</tr>
</tbody>
</table>

This book uses the following types of special text:

**NOTE**

Notes contain important information that you should consider.
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 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)

The MainView SYSPROG Services JITOKEN service, if given any of the elements in this 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 used. 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 253.
Expressions

Several MainView SYSPROG Services accept 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.

Expressions consist of one or more terms, separated by an operator. 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.

  Symbols are discussed in more detail under “Symbols” on page 19. MainView SYSPROG Services currently provides approximately 150 predefined symbols. You can also use the EQUATE service to define your own symbols. See “EQUATE” on page 221 for additional information.

- 6C? is a hexadecimal value (6C), followed by an indirection indicator (?). Indirection indicators are discussed in more detail under “Indirection indicators” on page 19.

- C0 is another hexadecimal value. 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 21.
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:

%  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.

?  A question mark indicates that the full-word value at the indicated address is to be treated as a 31-bit address value. In other words, the high-order bit in the full-word value is ignored, and the remaining 31 bits are substituted for the address.

!  An exclamation mark indicates that the content of the double word at the indicated address is to be treated as a 64-bit address value. In other words, the entire 64-bit value is substituted for the address.

>  A greater-than sign indicates that the contents of the three bytes at the indicated address are to be treated as an SWA address token. The 31-bit address represented by the SWA token is substituted for the address.

Indirection indicators do not need to be separated from each other. For example, 20??! is equivalent to 20?+0?+0!.

Symbols

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.

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.
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 do not usually 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

You can scale decimal numbers by appending the following suffix characters:

- **K** indicates that the preceding decimal value is in Kilobytes and should be scaled by multiplying it by 10 to the 3rd power
- **M** indicates that the preceding decimal value is in Megabytes and should be scaled by multiplying it by 10 to the 6th power
- **G** indicates that the preceding decimal value is in Gigabytes and should be scaled by multiplying it by 10 to the 9th power
- **T** indicates that the preceding decimal value is in Terabytes and should be scaled by multiplying it by 10 to the 12th power
- **P** indicates that the preceding decimal value is in Petabytes and should be scaled by multiplying it by 10 to the 15th power
- **Ki** indicates that the preceding decimal value is in Kilobytes and should be scaled by multiplying it by 2 to the 10th power
- **Mi** indicates that the preceding decimal value is in Megabytes and should be scaled by multiplying it by 2 to the 20th power
- **Gi** indicates that the preceding decimal value is in Gigabytes and should be scaled by multiplying it by 2 to the 30th power
- **Ti** indicates that the preceding decimal value is in Terabytes and should be scaled by multiplying it by 2 to the 40th power
- **Pi** indicates that the preceding decimal value is in Petabytes and should be scaled by multiplying it by 2 to the 50th power

**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`
- `200_`
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 312 for specific examples.

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. Consult the following table for the meaning of conventions used in these diagrams.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶▶</td>
<td>Double right arrowheads—the beginning of a syntax diagram.</td>
</tr>
<tr>
<td>▶</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>▶▶</td>
<td>Opposing arrowheads—the end of a syntax diagram.</td>
</tr>
<tr>
<td>▶▶keyword ▾▶</td>
<td>An item on the main line of the diagram is required.</td>
</tr>
<tr>
<td>▶▶literal ▾▶</td>
<td>A literal is represented in flat text.</td>
</tr>
<tr>
<td>▶▶variable ▾▶</td>
<td>A variable is represented in italic text.</td>
</tr>
<tr>
<td>▾▶keyword ▾▶</td>
<td>An item above the main line of the diagram is the default.</td>
</tr>
<tr>
<td>▾▶keyword ▾▶</td>
<td>An item underneath the main line of the diagram is optional.</td>
</tr>
</tbody>
</table>
Understanding MainView SYSPROG Services command syntax

The descriptions of services in Chapter 5, “Services,” use diagrams to illustrate command syntax. (See “Identifying address spaces” on page 17 for information about how to read syntax diagrams.) For example, the syntax for the CDE service is shown in the following diagram.

The diagram indicates that the CDE service accepts three positional parameters:

- *asi*—address space instance; see “Identifying address spaces” on page 17 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

---

**NOTE**

The examples in this section use the CDE service to demonstrate command syntax, but the syntax rules described here are applicable to all services.
When you invoke a service from the COMMAND line, in most cases you can use either a blank space or a comma to separate the service name and the parameters. (For clarity, the syntax diagrams in this book do not show commas between parameters.)

For example, both of these service commands are valid:

```
cde inventry
```
```
cde,inventry
```

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:

```
cde
```

This command displays CDE information about all modules in your own address space (because your own address space is the default for the `asi` parameter).

The CDE service also allows you to display information about a single module by specifying the `modname` parameter after the `asi` parameter:

```
cde inventry ieavtsdt
```

This command displays information about the module IEAVTSDT in the address space INVENTRY.

### Using commas as placeholders for default values

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. However, you must insert commas as placeholders for the `asi` parameter, as shown here:

```
cde,ieavtsdt
```

Most MainView SYSPROG Services parameters are positional; for example, in the CDE service, MainView SYSPROG Services would expect to find `asi` as the first parameter and the `modname` as the second parameter. If you type a value for `modname` without using commas as placeholders for `asi`, MainView SYSPROG Services interprets the module name as an address space name, and you are likely to get an error message.
Introducing MainView SYSPROG Services

This chapter 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.

This chapter includes the following topics:

MainView SYSPROG Services features .............................................. 26
Supported environments ................................................................. 27
How MainView SYSPROG Services works with other products ............. 28
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.

  For information about implementing SAF security for MainView SYSPROG Services, see the *MainView Security Reference Manual*. 
MainView SYSPROG Services operates as an authorized program within the MainView Product Address Space (PAS), as a TSO command within a TSO address space, or as a batch job or Started Task. You can use MainView SYSPROG Services in the operation modes that best suit your needs. Table 1 lists each of the available modes and gives a brief description of each.

Table 1 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. <strong>Note:</strong> SYSPROG must be defined as an authorized command.</td>
</tr>
<tr>
<td>Started Task</td>
<td>You can also use the MODIFY command to enter MainView SYSPROG Services commands from a system console.</td>
</tr>
<tr>
<td>Batch Job</td>
<td>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 <strong>PARM=BATCH</strong> on the EXEC statement.</td>
</tr>
</tbody>
</table>

For more information about the operation modes, see “Independent operations” on page 54.
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 Chapter 5, “Services,” for a complete description of the available services.

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

Exception Monitor samplers

The Advanced Early Warning (AEW) 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 AEW. The manner in which warning messages are displayed depends upon the environment within which AEW 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.

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 ALRTRCV=YES specified in BBISPxx can monitor and respond to various warning messages. Specify AEWALERT=YES in BBPARM member $INSYS0 to activate this facility. See “$INSYS0” on page 34.
Chapter 2
Controlling the MainView SYSPROG Services environment

This chapter describes the various parameter library members and how you can use them to control various aspects of SYSPROG Services. The following topics are covered:

How MainView SYSPROG Services locates the parameter library members . . . . . . . 29
PARMLIST: parameter member processing report . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
Use of symbols within members . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
Parameter library members . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

- $$INEXCL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
- $$INNWTO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
- $$INSYS0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
- $$RPJOBS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
- $$RPTEXT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
- $$XENQ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
- $$XRES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
- PWSCPMxx . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

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 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 partitioned data sets allocated to DDNAME BBPARM.
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.

Use of 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 AEW samplers for each LPAR by using a single parameter library that is shared across multiple LPARS.

Parameter library members

Table 2 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 page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$INEXCL</td>
<td>specifies which job names are excluded from exception reporting by specified Exception Monitor services</td>
<td>31</td>
</tr>
<tr>
<td>$$INNWTO</td>
<td>restricts MainView SYSPROG Services command entry to the Modify mode</td>
<td>33</td>
</tr>
<tr>
<td>$$INSYS0</td>
<td>defines runtime parameters for MainView SYSPROG Services</td>
<td>34</td>
</tr>
<tr>
<td>$$RPJOBS</td>
<td>exempts jobs with constantly outstanding WTORs from the MainView SYSPROG Services outstanding reply scan</td>
<td>37</td>
</tr>
</tbody>
</table>
The following sections describe the BBPARM library members, including syntax rules and examples.

### Syntax rules

The format is

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

The parameters are defined as follows:

- **samplerName**
  - name of the Exception Monitor sampler that is to exclude selected job names from exception reporting
  - The sampler names that you can specify are 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.

- **jobname1–jobname10**
  - names of up to ten jobs that are to be excluded from exception reporting
$INEXCL

**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**
This member suppresses the WTOR message AMT000A ENTER SYSPROG COMMAND, issued when SYSPROG Services is running as a batch job or a Started Task.

When you run MainView SYSPROG Services as a batch job or a Started Task, you can enter MainView SYSPROG Services commands in two ways:

- by using the MVS MODIFY command
- by replying to the WTOR message AMT000I ENTER SYSPROG COMMAND

If you want to use only the MODIFY command to enter MainView SYSPROG Services commands, you can suppress AMT000I ENTER SYSPROG COMMAND WTOR messages with member $$INNWTO by placing member $$INNWTO in the BBPARM data set; no records need to be present.

In addition, you can suppress the AMT000I ENTER SYSPROG COMMAND WTOR message by specifying a z/OS subsystem command ID character to identify MainView SYSPROG Services commands. See the description of the CMDID keyword in member $$INSYS0 (on page 35) for more information.
This member defines runtime parameters for MainView SYSPROG Services. MainView SYSPROG Services reads this member when it initializes.

**Syntax rules**

Table 3 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>
<thead>
<tr>
<th><strong>Keyword=value</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AEWALERT=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AEWNOROLL=0</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The AEWNOROLL value can also be modified by using the SETOP service.</td>
</tr>
<tr>
<td>Keyword=value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEWSTART=00</td>
<td>xx</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>CAPS=OFF</td>
<td>ON</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDID=char</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMP=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ECHO=ECHO</td>
<td>NOECHO</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3  Keywords for member $SINSYS0 (part 3 of 3)

<table>
<thead>
<tr>
<th>Keyword=value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOSMSG=NO</td>
<td>YES</td>
</tr>
<tr>
<td>- NO—does not issue an END OF SERVICE message at the completion of every service (default)</td>
<td></td>
</tr>
<tr>
<td>- YES—issues an END OF SERVICE message at the completion of every service</td>
<td></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. Specify a value between 0 and 15. The default value is 3 seconds.</td>
</tr>
<tr>
<td>LCLAEW=NORMAL</td>
<td>CONSOLE</td>
</tr>
<tr>
<td>LCLAEWTS=NO</td>
<td>YES</td>
</tr>
<tr>
<td>- No- does not time stamp Exception Monitor sampler messages (default)</td>
<td></td>
</tr>
<tr>
<td>- Yes- time stamps Exception Monitor sampler messages</td>
<td></td>
</tr>
<tr>
<td>MSGID=YES</td>
<td>NO</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.
This member exempts jobs with constantly outstanding WTORs that are displayed by the MainView SYSPROG Services REPLIES service and the REP sampler.

See also the description for $$RPTEXT on page 38.

**Syntax rules**

The following syntax rules apply:

- Specify one job name per entry.
- 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:

```none
IMSDC
```
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"
```
This member specifies major/minor name pairs to be excluded from the ENQUEUEES 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/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.

**Example 2**

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

**Example 3**

All conflicts with a major name of ABC will be excluded, regardless of the minor name.
This member specifies volume serial numbers that are to be excluded from reporting by the RESERVES service and the AEW 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 AEW 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, PWSCPM9A). 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 Monitors listed in member PWSCPM00 are invoked when you begin Exception Monitor sampling.

**Example**

The following example is of a member called PWSCPMEX:

```
INT 60
ENQ 5,10
PAG 4,50
A1OR 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 (**A1OR 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.
Chapter 3 Using MainView SYSPROG Services

This chapter describes the methods by which you can access MainView SYSPROG Services, either through MainView for z/OS or as a stand-alone product. It includes the following topics:

- Accessing MainView SYSPROG Services ........................................... 45
- Executing MainView SYSPROG Services from an operator console .......... 53
- Independent operations ..................................................................... 54
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.
- 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

Figure 1 shows the MainView Selection Menu panel.

Figure 1  MainView Selection Menu panel

<table>
<thead>
<tr>
<th>OPTION</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Parameters and Options</td>
</tr>
<tr>
<td>E</td>
<td>Alerts and Alarms</td>
</tr>
<tr>
<td>P</td>
<td>PLEX Management (PLEXMGR)</td>
</tr>
<tr>
<td>U</td>
<td>Utilities, Tools, and Messages</td>
</tr>
</tbody>
</table>

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. 2002

From this panel, select option 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 46.
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** shows an example.
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 Figure 4, or by typing EZMSPROG on the COMMAND line.

The primary MainView SYSPROG Services menu is the SYSPROG Easy Menu (EZMSPROG), shown in Figure 3 on page 46.
SYSPROG Services Fast Menu

The SYSPROG Services Fast Menu, an option on SYSPROG Easy Menu and shown in Figure 5, 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    SYSPROG Fast Menu (EZMFPROG)
Accessing Services from the SYSPROG Services Menu panel

An alphabetical list of all available services is displayed on the SYSPROG Services Menu panel, as shown in Figure 6.

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 6. 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>Command</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>DVIEW</td>
<td>View DASD cchhr/dsn/dscb/extent/files/label</td>
<td></td>
</tr>
<tr>
<td>EDTINFO</td>
<td>Display EDT information</td>
<td></td>
</tr>
<tr>
<td>ENQUEUES</td>
<td>Displays enqueue conflicts</td>
<td></td>
</tr>
<tr>
<td>EQUATE</td>
<td>Define symbols for use with the dump service</td>
<td></td>
</tr>
</tbody>
</table>

Scroll down to display the rest of the services. Each service is described in Chapter 5, “Services.”
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 summarizes the functions of the MainView SYSPROG Services commands L, S, and E; subsequent sections in this chapter provide more details.

<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 L serviceName.</td>
<td>“Locating a service”</td>
</tr>
<tr>
<td>find out more about a service, or change its parameters</td>
<td>Use the S line command.</td>
<td>“Finding out more about a service” on page 51</td>
</tr>
<tr>
<td>execute a service</td>
<td>Choose one:</td>
<td>“Executing a service” on page 52</td>
</tr>
<tr>
<td></td>
<td>- Use the E line command (uses defaults).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- On the COMMAND line, type the name of the service and its parameters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Use the S line command to select the panel so that you can specify the parameters, and then press Enter.</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 shows the Service Information panel for the DVIEW service.

**Figure 7  Service Information panel for DVIEW**

```
 COMMAND ===>

 The DVIEW service displays data blocks on any online DASD volume.

 OPTION ===> (CCHHR, DSN, DSCB, EXTENTS, FILES or LABEL)
 VOLSER ===> (Volume serial number of DASD device to view)

 Required for CCHHR:
 ADDRESS ===> (Absolute track address, format cccchhhhr)

 Required for DSN, DSCB or EXTENTS; optional partial DSN for FILES:
 DSNAM E ===>>

 Optional for DSN:
 SKIP ===> (Number of blocks to skip--default 0)
 COUNT ===> (Number of blocks to display--default 1)

 Press ENTER to execute service
 END to cancel request
 Press HELP for more information.
```

To execute the service from the Service Information panel, type values in the parameter input fields, if desired, and press Enter. Otherwise, press the END key to exit the panel.
Accessing Services from the SYSPROG Services Menu panel

**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* shows the SYSPROG Services Output panel after the DVIEW service is executed.

*Figure 8* SYSPROG Services Output panel for DVIEW

```
------------------------------SYSPROG Services Output------------------ Row 1 of 4
COMMAND ===> DVIEW,DSN,BAB321,BMVSPS.CMF.JCL   SCROLL ===> PAGE
TARGET - CXTSTW

15:00:54 CMD=DVIEW,DSN,BAB321,BMVSPS.CMF.JCL

AMTVD021 DISK VIEW SERVICE
AMTVD031 VOL - BAB321   BLOCK CCHHR - 0539000001
AMTVD061 CSW - 0007D400 0000FEF7  BLKSIZE - 264
AMTVD111 ECB - 40000045  SENSE - 0000
AMTVD071 0000 5BC2F05F  E4E44040  00FE5B5B  C9D5D3C7  * .B05UU  ...INLG+
AMTVD071 0010 C4D70004  160F0104  00570101  290F0102  *DP............*
AMTVD071 0020 04F1410  00050005  0003C2D4  E5C4E2D7  * ........BMVDSP*
AMTVD071 0030 F5A40404  5BC14040  40404040  004180F1  *5 .A ....*
AMTVD071 0040 1010048  0102291F  0102291F  1202002A  * ........*
AMTVD071 0050 02A00000  C204E5E2  D7E2F240  404058C1  * ....BMVSPS2 .A*
AMTVD071 0060 C3C3C5E2  E2400004  1A0F0101  00180102  *CESS ........*
AMTVD071 0070 071F0102  080F1550  0007000B  0002C2D4  * ..........BM*
AMTVD071 0080 D5E2DE7E  F4404040  5BC1D5D6  E3C5E240  *VSPS4 .ANOTES *
AMTVD071 0090 0041C0F  01010008  0101107F  0101107F  * ........*
AMTVD071 00A0 11080014  00010000  C2DA5E5E2  D7E2F340  * ........BMVSPS3 *
```

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

You can invoke MainView SYSPROG Services from an z/OS operator console by using the MODIFY (F) command or a predefined Command Identification (CMDID) character.

Using the MODIFY (F) command

Specify the MODIFY (F) command in this format:

```
F pasname,R=sysprogService
```

where `pasname` is the name of the PAS and `sysprogService` is the name of the service that you want to invoke, along with any parameters that the service might require.

Using a CMDID character

You can define a CMDID character in the JCL for the z/OS PAS.

If you have already defined a CMDID character in the z/OS PAS JCL, you can invoke MainView SYSPROG Services by preceding the service name with two of the defined CMDID characters. The pair of characters tells the system to invoke MainView SYSPROG Services, as R= does in the previous example.

Type the command in this format:

```
**sysprogService
```

where `*` is any valid CMDID character.

For example, if you specified `CMDID=#` in the z/OS PAS JCL, you could type `##INFO` at the console to invoke the INFO service.

Valid CMDID characters

The following characters are valid CMDID characters. They can be specified as a single character, a single character in quotation marks, or a hexadecimal equivalent (two digits).

```
c . < ( + | & ! * ) _ - / % _ > ? : # @ ' = "
```
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 in response to an outstanding WTOR prompt or by using an operator MODIFY command or a command identification character. The specific method is determined by options specified in BBPARM member $$INSYS0 and the presence (or absence) of member $$INNWTO. See Chapter 2, “Controlling the MainView SYSPROG Services environment,” for details.

To use MainView SYSPROG Services as a started task, create a PROCLIB member, patterning it after the sample shown in Figure 9; then place it in a cataloged procedure data set.

Figure 9  Sample JCL for MainView SYSPROG Services

```
//SYSPROC EXEC PGM=SYSPROC,DPRTY=(15,15)
//STEPLIB DD DISP=SHR,DSN=????????.BBLINK [1]
//BBPARM DD DISP=SHR,DSN=????????.UBBPARM [2]
//LOG DD SYSOUT=A [3]
```

Legend

1. The STEPLIB library is not required if the BBLINK data set is in the linklist. If you include a STEPLIB DD statement, the data set must be APF authorized.

2. The BBPARM library defines a partitioned data set that contains optional initialization parameters, Early Warning parameters, and members that are input to the EXEC service.

3. (optional) LOG defines the default output data set for the LOG service.

4. (optional) TRACK defines the default output data set for the TRACK service.
To start MainView SYSPROG Services, type

```
S SYSPROG.id
```

where *id* is the started task identifier.

BBSAMP member SYSPROGJ contains a sample procedure for executing MainView SYSPROG Services as a Started Task, which you can copy to a cataloged procedure data set. This member also contains an explanation of each DD statement that is contained in the procedure.

---

**NOTE**

You can change the MainView SYSPROG Services library names to correspond to the standards at your site.

---

**Executing Services in Started Task mode**

If you are running MainView SYSPROG Services in Started Task mode, you can enter commands by using the MODIFY (F) command. For example, to invoke the CPU service, type

```
F jobname|ID,CPU
```

Using MODIFY (F) to enter commands in Started Task mode is recommended for these reasons:

- The current outstanding reply number does not have to be found; thus, you can enter the command faster and with less chance of syntax error.
- The service returns output only to the console from which the service was invoked.

A console must have SYS or ALL authority before commands can be executed through MODIFY. The message **CONSOLE NOT AUTHORIZED** is displayed if the authority has not been assigned.

After you use MODIFY (F) to enter a command, MainView SYSPROG Services responds with

```
AMTOODI 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**

In order for MainView SYSPROG Services to run in the TSO environment, you must set the authorization through AutoCustomization or through manual customization.

In AutoCustomization, select the step with the description Authorize SYSPROG to run in a TSO environment. Refer to the MainView Customization Reference for information about how to complete this step manually.
MainView SYSPROG Services as a batch job

To execute a series of services as a batch job, place MainView SYSPROG Services commands in the JCL shown in Figure 10 and submit the job for processing. Output appears in the JES held output queue.

Figure 10 Sample batch job for MainView SYSPROG Services

```
//SYSPROG EXEC PGM=SYSPROG,REGION=1024K,DRTY=(15,15),PARM='BATCH'
//STEPLIB DD DSN=hilevel.BBLINK,DISP=SHR
//BMCPWMD DD DISP=SHR,DSN=??????????.BMCPSWD
//BBPARM DD DSN=hilevel.BBPARM,DISP=SHR
//LOG DD SYSOUT=A,DCB=BLKSIZE=141 [1]
//TRACK DD SYSOUT=A,DCB=BLKSIZE=141 [1]
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
CSA [2]
ASM,MAP [2]
END [3]
/*
```

Legend

1. (optional)

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

3. Specify END as the last statement
This chapter provides detailed information about the Exception Monitor samplers, including proper syntax, required and optional parameters, and usage examples. It includes the following topics:

<table>
<thead>
<tr>
<th>Sampler name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPU</td>
<td>62</td>
</tr>
<tr>
<td>AIO</td>
<td>64</td>
</tr>
<tr>
<td>AIOR</td>
<td>66</td>
</tr>
<tr>
<td>APAG</td>
<td>67</td>
</tr>
<tr>
<td>APGR</td>
<td>69</td>
</tr>
<tr>
<td>ASRV</td>
<td>71</td>
</tr>
<tr>
<td>ASSR</td>
<td>73</td>
</tr>
<tr>
<td>ASVT</td>
<td>75</td>
</tr>
<tr>
<td>BCPU</td>
<td>77</td>
</tr>
<tr>
<td>BIO</td>
<td>78</td>
</tr>
<tr>
<td>BPAG</td>
<td>80</td>
</tr>
<tr>
<td>BSRV</td>
<td>82</td>
</tr>
<tr>
<td>CADP</td>
<td>84</td>
</tr>
<tr>
<td>CHA</td>
<td>85</td>
</tr>
<tr>
<td>CPU</td>
<td>87</td>
</tr>
<tr>
<td>CQD</td>
<td>89</td>
</tr>
<tr>
<td>CSA</td>
<td>90</td>
</tr>
<tr>
<td>CSMJ</td>
<td>93</td>
</tr>
<tr>
<td>DEV</td>
<td>95</td>
</tr>
<tr>
<td>DIOQ</td>
<td>97</td>
</tr>
<tr>
<td>DNR</td>
<td>99</td>
</tr>
<tr>
<td>DVTN</td>
<td>100</td>
</tr>
<tr>
<td>ENQ</td>
<td>102</td>
</tr>
<tr>
<td>FIX</td>
<td>103</td>
</tr>
<tr>
<td>INT</td>
<td>105</td>
</tr>
<tr>
<td>JET</td>
<td>106</td>
</tr>
<tr>
<td>JSU</td>
<td>108</td>
</tr>
<tr>
<td>JVR</td>
<td>110</td>
</tr>
<tr>
<td>MTP</td>
<td>111</td>
</tr>
<tr>
<td>NRQ</td>
<td>112</td>
</tr>
<tr>
<td>NRQP</td>
<td>113</td>
</tr>
<tr>
<td>OUT</td>
<td>115</td>
</tr>
<tr>
<td>PAG</td>
<td>117</td>
</tr>
<tr>
<td>PCPU</td>
<td>118</td>
</tr>
<tr>
<td>PER</td>
<td>120</td>
</tr>
<tr>
<td>PGQ</td>
<td>122</td>
</tr>
<tr>
<td>REP</td>
<td>123</td>
</tr>
<tr>
<td>RES</td>
<td>124</td>
</tr>
<tr>
<td>RSU</td>
<td>126</td>
</tr>
<tr>
<td>SMF</td>
<td>127</td>
</tr>
<tr>
<td>SPA</td>
<td>128</td>
</tr>
<tr>
<td>SPLW</td>
<td>130</td>
</tr>
<tr>
<td>SRM</td>
<td>131</td>
</tr>
<tr>
<td>SWP</td>
<td>132</td>
</tr>
<tr>
<td>TCPU</td>
<td>133</td>
</tr>
<tr>
<td>TIO</td>
<td>134</td>
</tr>
<tr>
<td>TPAG</td>
<td>136</td>
</tr>
<tr>
<td>TSRV</td>
<td>137</td>
</tr>
<tr>
<td>Sampler name</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>WDLY</td>
<td>139</td>
</tr>
<tr>
<td>WOBJ</td>
<td>141</td>
</tr>
<tr>
<td>WRT</td>
<td>142</td>
</tr>
<tr>
<td>WTO</td>
<td>143</td>
</tr>
<tr>
<td>WTOR</td>
<td>144</td>
</tr>
<tr>
<td>XCPU</td>
<td>146</td>
</tr>
</tbody>
</table>
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. Messages can be sent to the system console in addition to the WARN view.

To access the Exception Monitors

1. Access MainView SYSPROG Services from the z/OS and USS Solutions panel (or, on the COMMAND line, type SYSPROG).

2. Select the WARNING service.

Each sampler monitors a different aspect of system performance. You control the Exception Monitor by selecting samplers that are critical to your site, setting thresholds for them and then building a control statement member with those samplers.

For information about how to create a control statement member, see the MainView Customization Reference.
ACPU

The Address Space CPU (ACPU) sampler sends a warning message to the operator when an address space uses more than a specified percentage of CPU time during the reporting interval. You can suppress the message if a given threshold for total system CPU utilization 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 31.

**Syntax**

```
ACPU intervals name percent type total cpu%
```

The parameters are defined as follows:

- **intervals** specifies 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.

- **percent** is the CPU busy percentage for which an address space must be responsible before an exception message is issued for that address space

- **type** is the type of address space to be monitored, where
**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the system CPU-busy percentage exceeds 50 percent over the preceding two minutes:

```
acpu 4.*,35,a,50
```

Also, a warning is issued for any address space that has been active 35 percent of the total CPU-busy time.

**Warning message**

The following warning message is issued by the ACPU sampler:

```
PWSACP94 *WARNING* CPU usage is percent% for type name
```

Message PWSACP94 is displayed when the ACPU warning sampler detects that a job, task, or TSO user is using more than a specified percentage of CPU time during the reporting interval, but only if the system is busier than the threshold that was set when the sampler was defined.

The variable fields in the message are:

- `percent` is the percentage of cpu time that the address space used
- `type` is the type of address space (Job, STC, TSU, or All)
- `name` is the name of the address space exceeding the limit
AIO

The Address Space I/O (AIO) sampler sends a warning message to the operator 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 31.

Syntax

```
AIO (intervals name percent A type 25 i/o rate)
```

The parameters are defined as follows:

- **intervals**: specifies 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.

- **percent**: 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
  - If this value is zero, no warning messages will be issued.

- **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 the system I/O rate exceeds 50 EXCPs-per-second over the preceding two minutes:

\[ \text{aio 4,*,35,a,50} \]

Also, a warning is issued for any address space doing more than 35 percent of the I/O.

Warning messages

The following warning messages are issued by the AIO sampler:

\[ \text{PWSAI090 *WARNING* TOTAL SYSTEM I/O RATE IS pppppp EXCPS/SEC} \]

Message PWSAI000 indicates that the system-I/O rate has exceeded the warning threshold.

\[ \text{PWSAI092 *WARNING* I/O RATE FOR ttt nnnnnnn IS xxx% OF TOTAL} \]

Message PWSAI001 indicates that the specified address space has exceeded the warning threshold for percentage of system I/O, where

\[
\begin{align*}
\text{ttt} & \quad \text{is STC, TSU, or JOB} \\
\text{nnnnnnnn} & \quad \text{is the address space name} \\
\text{xxx} & \quad \text{is the percent of the total system I/O rate}
\end{align*}
\]
The Address Space I/O Rate (AIOR) sampler sends a warning message to the operator 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 31.

**Syntax**

```
AIOR interval name i/o rate type
```

The parameters are defined as follows:

- **interval** is the number of report intervals between warning messages
- **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.
- **i/o 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:

aior 4,*,25,a

Warning message

The following warning message is issued by the AIOR sampler:

PWSAIR91 *WARNING* I/O RATE FOR ttt nnnnnnnn IS excp EXCPs/SEC

Message PWSAIR01 indicates that the specified address space has exceeded the warning threshold for I/O rate, where

- \( ttt \) is STC, TSU, or JOB
- \( nnnnnnnn \) is the address space name
- \( excp \) is the EXCPs issued by the address space during the interval

APAG

The Address Space Paging (APAG) sampler sends a warning message to the operator 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 31.

Syntax

The parameters are defined as follows:
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.
Message PWSAPAG1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

- **ttt** is STC, TSU, or JOB
- **nnnnnnnn** is the address space name
- **xxx** is the percentage of the total system paging rate

The Address Space Paging Rate (APGR) sampler sends a warning message to the operator 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 31.

**Syntax**

```
APGR intervals name >10 paging rate type
```

The parameters are defined as follows:

- **intervals** specifies the length in intervals of the sample period
  
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.
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 nnnnnnnn IS ppp PAGES/SEC
```

Message PWSAPGR1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

- `ttt` is STC, TSU, or JOB
- `nnnnnnnn` is the address space name
- `ppp` is the number of pages per second
The Address Space Service Rate Percentage (ASRV) sampler sends a warning message to the operator 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.

**NOTE**
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 31.

**Syntax**

```
ASRV intervals name-percent type measure service rate
```

The parameters are defined as follows:

- **intervals** specifies 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.

- **percent** 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

- **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 whenever an address space being monitored uses more than the 35 percent of total SRM service components during a given reporting interval:

```plaintext
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 messages are issued by the ASRV sampler:

PWSASRV0 *WARNING* ADDR SPACE mmm SERVICE RATE IS sssss SU/SEC

Message PWSASRV0 indicates that the system service-rate has exceeded the warning threshold, where

- **mmm** is the SRM service component being monitored
- **sssss** is the number of service units

PWSASRV1 *WARNING* mmm SERVICE RATE FOR ttt jjjjjjjjj IS ppp% OF SYSTEM TOTAL

Message PWSASRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

- **mmm** is the SRM service component being monitored
- **ttt** is STC, TSU, or JOB
The Address Space Service Unit Rate (ASSR) sampler sends a warning message to the operator when an address space uses SRM service components at a rate that exceeds the defined threshold during a reporting interval. The message can be suppressed if a given total system-service rate threshold is not reached during a reporting interval.

### Syntax

The parameters are defined as follows:

- **interval**
  is the number of report intervals between warning messages

- **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 SRM component service rate for an address space that must be reached before exception messages are issued

  The default is a system service rate greater than zero SUs per second.

- **type**
  is the type of address space to be monitored, where

---

**NOTE**

You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 31.
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 jjjjjjj IS rrrrr SU/SEC`

Message PWSASSR1 indicates that the address-space service-unit rate has exceeded the warning threshold, where

- `mmm` is the SRM service component being monitored
- `ttt` is STC, TSU, or JOB
- `jjjjjjjj` is the address space name
- `rrrrr` is the number of service units
The ASVT sampler displays 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**

The parameters are defined as follows:

- **intervals** specifies the length of the report period in intervals; the default is 30 seconds (one interval)

- **avq** is the threshold percentage for the Available Queue

  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.

- **rplq** is the threshold percentage for the Non-reusable Replacement Queue

  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.
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 the system was IPL’d.

### 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:

  \[
  \left( \frac{\text{reserved-slots-available}}{\text{remaining-reserved-slots}} \right) \times 100
  \]
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:

\[
\frac{\text{reserved-STC-slots}}{\text{remaining-STC-slots}} \times 100
\]

The Batch CPU (BCPU) sampler sends a warning message to the operator when a batch address space uses 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 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 31.

**Syntax**

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **percent** 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
- **rptno** is the maximum number of address spaces to be listed whenever a CPU overload condition is detected; the default is 5
- **total cpu %** 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
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 jjjjjjjj
```

Message PWSBCPU1 indicates that the specified address space has exceeded the warning threshold for percentage of system-CPU busy time, where

- `ppp` is the address space busy percentage
- `jjjjjjjj` is the address space name

BIO

The Batch I/O (BIO) sampler sends a warning message to the operator 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 31.
Syntax

The parameters are defined as follows:

- **intervals**: specifies the length in intervals of the sample period; the default sample period is 30 seconds (one interval)
- **percent**: 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
- **rptno**: is the maximum number of batch address spaces to be listed whenever a batch I/O overload condition is detected
- **i/o rate**: 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

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 messages are issued by the BIO sampler:

PWSBIO94 *WARNING* TOTAL SYSTEM I/O RATE IS xxx EXCPS/SEC

Message PWSBIO94 indicates that the system-I/O rate has exceeded the warning threshold.

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

nnnnnnnn is the address space name

xxx is the percent of the total system I/O rate

BPAG

The Batch Paging (BPAG) sampler sends a warning message to the operator 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 31.

Syntax

The parameters are defined as follows:
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:

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

nnnnnnnn is the address space name
xxx is the percent of total system paging rate
The Batch Service (BSRV) sampler sends a warning message to the operator 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.

NOTE
You can suppress reporting for specific address spaces by including their names in member BBPARM $$INEXCL. For details, see “$$INEXCL” on page 31.

Syntax

The parameters are defined as follows:

- **intervals** specifies the length in intervals of the sample period; the default sample period is 30 seconds (one interval)
- **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** specifies the SRM service measure to be monitored:
  - C CPU service units
  - I I/O service units
  - M MSO service units
  - T 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
- **service rate** 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 messages are issued by the BSRV sampler:

```
PWSBSRV0 *WARNING* ADDR SPACE mmm SERVICE RATE IS sssss SU/SEC
```

Message PWSBSRV0 indicates that the system paging rate has exceeded the warning threshold, where

- `mmm` is the SRM service component being monitored
- `sssss` is the number of service units per second

```
PWSBSRV1 *WARNING* mmm SERVICE RATE FOR JOB jjjjjjj IS ppp% OF SYSTEM TOTAL
```

Message PWSBSRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

- `mmm` is the SRM service component being monitored
- `jjjjjjjj` is the address space name
- `ppp` is the address space SRM component use as a percentage of the system total
The Common Area Dataspace (CAD) Percentage sampler determines the percentage of the permitted Common Area Dataspaces that have been created and issues a warning message when the user-specified percentage has been exceeded.

**Syntax**

The parameters are defined as follows:

- **rptint** specifies the length of the sample period in intervals
  The default sample period of 30 seconds (one interval) is the percentage utilization of CSA that causes a warning message to be issued

- **thresh** specifies 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.

**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

- $percent$ current percentage of CAD dataspaces used
- $used$ number CAD dataspaces used
- $lost$ number of CAD dataspaces lost

**CHA**

The Channel (CHA) sampler monitors channel-path usage and sends a warning message to the operator when a channel path exceeds the defined utilization-percentage threshold.

**Syntax**

```
CHASHA  1
intervals  threshold
chpid1  chpid2
```

The parameters are defined as follows:

- $intervals$ specifies the length in intervals of the sample period; the default sample period is 30 seconds (one interval)
- $threshold$ is the average utilization high threshold (0-100 percent)
- $chpid1$ is the lowest path number in the range of channel paths to be monitored; must be two digits
- $chpid2$ is the highest path number in the range of channel paths to be monitored; must be two digits

**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

- `xx` is the physical channel number (0 to 15)
- `y` is the CPU number (0 or 1)
- `zz` is the percent utilization
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 31.

**Syntax**

The parameters are defined as follows:

- **intervals** specifies the length in intervals of the sample period; the default sample period is 30 seconds (one interval)

- **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.

- **TSOLIMIT=nn** is the percent CPU utilization by TSO users that causes a warning message to be issued
  
  Specify two digits to represent the percentage; the valid range is 0 to 99.

- **JOBLIMIT=nn** is the percent CPU utilization by individual jobs, TSO users, or Started Tasks that causes a warning message to be issued
  
  Specify two digits to represent the percentage; the valid range is 0 to 99.
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

type is the type of user (TSO, JOB, STC)
percent is the percentage of the current CPU utilization for the specified type
The CPU Queue Depth (CQD) sampler monitors the CPU active task queue. It also sends a warning message to the operator when the average queue depth of active tasks (both those in memory and those ready for dispatching) exceeds a threshold you specify.

**Syntax**

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `depth` is the threshold count of active tasks waiting

**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 active tasks waiting.
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**

```
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>interval</td>
</tr>
<tr>
<td>----------</td>
</tr>
</tbody>
</table>
```

The parameters are defined as follows:

- `intervals`: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
**threshold** is the percent utilization of CSA that causes a warning message to be issued.

There are two thresholds:

1. csapct—when the amount of allocated CSA, as determined by the chosen method, reaches or exceeds the specified value.

2. ecsapct—when the amount of allocated ECSA, as determined by the chosen method, reaches or exceeds the specified value.

If omitted, the CSA threshold is used.

**method** specifies 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 assumed to be 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:

```
PWSCSA01 *WARNING* CSA usage is xxx%; remaining free space is yyyy yyyy?
```

Message PWSCSA01 indicates that CSA usage exceeds the specified threshold, where
xxx is the percentage of CSA used

The percentage used includes fragmented free storage.

yyyyyyyy 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.
The CSMJ sampler monitors the percentage of common storage use by job names and sends a warning message to the operator for each job that meets or exceeds the specified thresholds. This information allows the operator 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.

Global thresholds apply to all jobs except *MASTER* and *SYSTEM*. The default global thresholds are 3% for CSA and SQA, and 2% for ECSA and ESQA.

The master address space (*MASTER*) and storage attributed to the system (*SYSTEM*) will be checked only when specified limits are provided for these names.

You can specify threshold overrides for specific job names. Address spaces that can allocate large amounts of common storage (such as JES2, IBM IMS, and IBM DB2®) should have their job names specified and their threshold values raised accordingly, which prevents the issuance of multiple warning messages.

**NOTE**

You can suppress reporting for specific address spaces by including their names in member BBPARM $INEXCL. For details, see “$INEXCL” on page 31.

### Syntax

```
CSMJ
```

```
intervals
```

```
t1 t2 t3 t4 jobn
```

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **t1** is the percent allocation of CSA by any job name that causes a warning message to be issued
- **t2** is the percent allocation of ECSA by any job name that causes a warning message to be issued
- **t3** is the percent allocation of SQA by any job name that causes a warning message to be issued
- **t4** is the percent allocation of ESQA by any job name that causes a warning message to be issued
**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message at 30-second intervals whenever any job has allocated more than the global thresholds, or for job name *MASTER*, whenever it has allocated more than four percent of CSA or SQA and more than three percent of ECSA or ESQA:

```
csmj 3,2,3,2, *master*,4,3,4,3
```

---

**Note**

Control statements ending with a comma (,) indicate that the statement is continued on the next input line. Continuation lines must be indented if the job name has an asterisk as the first character.

---

**Warning messages**

The following warning messages are issued by the CSMJ sampler:

**PWSCSM01** *WARNING*  xxxxxxxx  yyy  HAS ALLOCATED  nn%  OF  zzzz

Message PWSCSM01 indicates that job name  xxxxxxxx  in ASID  yyy  has allocated  nn  percent of available common storage. The common storage area (CSA, ECSA, SQA, or ESQA) is indicated by  zzzz.

**PWSCSM02** *WARNING* FAILURE EXTRACTING CS MONITOR DATA

Message PWSCSM02 indicates that an error has occurred in the COMMON STORAGE MONITOR component of BBXS. Contact BMC Software Customer Support.

**PWSCSM03** *WARNING* CS MONITOR IS NOT ACTIVE

Message PWSCSM03 indicates that COMMON STORAGE MONITOR must be active to use the CSMJ sampler. See the  MainView for z/OS User Guide and Reference  for information about starting the COMMON STORAGE MONITOR.

**PWSCSM04** *WARNING* BBXS LEVEL DOES NOT SUPPORT CS MONITOR
Message PWSCSM04 indicates that the active BBX subsystem is not at the required level to support COMMON STORAGE MONITOR. Ensure that current maintenance has been applied to FMID BBBBX16. An IPL or execution of the BBXSINIT batch job is required to activate the new level of BBXS after applying maintenance.

**PWSCSM05 *WARNING* BBXS IS NOT ACTIVE**

Message PWSCSM05 indicates that BBXS must be active to use the CSMJ sampler. BBXS is started automatically when MainView for z/OS is activated. Contact BMC Software Customer Support.

**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 [intervals threshold dvn] [dvn1-dvn2]
```

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **threshold** is the threshold percentage for device utilization
  
  A warning message is issued for each device in the range whose percent busy exceeds this threshold.
- **dvn** is a device number
  
  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.
- **dvn1-dvn2** is a range of device numbers; dvn1 is the lowest device number and dvn2 is the highest device number.
**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.
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**

![Diagram of DIOQ parameters]

The parameters are defined as follows:

- **intervals**: is the number of intervals between report periods; the default report period is 30 seconds (one interval)
- **depth**: 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.

**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:

```
PWSDIOQ0 *WARNING* Q=qqq nnnn a aa volser s mmm jobname
```

Message PWSDIOQ0 indicates that the I/O Queue depth has exceeded the warning threshold, where

- **qqq**: is the I/O queue depth for the DASD device
- **nnnn**: is the device type
aaa is the device address
volser is the volume serial number
s 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.

mmm is the mount attribute for the volume
jobname is the job name for the job with I/O in progress at the time that the device was checked
The Device Not Ready (DNR) sampler monitors tape and DASD devices online but not ready and notifies the operator when online devices have been not ready for more than a specified interval.

**Syntax**

```
DNR intervals notready
```

The parameters are defined as follows:

- `intervals`: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `notready`: is the number of minutes a device must remain not ready for a warning message to be issued

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals for each device that has been online but not ready for over two minutes:

```
dnr 4,2
```

**Warning message**

The following warning message is issued by the DNR sampler:

```
PWSNDNR01 *WARNING* DEVICE nnn (vvvvvv) NOT READY FOR xx MIN
```

Message PWSNDNR01 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.
The DVTN sampler displays deviation from a list of jobs that must be active during specified time intervals and notifies you when a job that should be active is not. This list of jobs resides in a CONFIGxx member of SYS1.PARMLIB.

**Syntax**

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **00** is the default suffix of the CONFIGxx member
- **xx** is the suffix of the CONFIGxx member in SYS1.PARMLIB that is to be used

The entries in the CONFIGxx member must begin in column 1. The format of the entries is

`*/NAME=jobname TIME=hhmm-hhmm ACT=command`

The parameters are defined as follows:

- **jobname** is the job name (or Started Task ID for Started Tasks)
- **hhmm-hhmm** are the beginning and ending times for the interval that the specified job must be active; a beginning value of four zeros (0000-hhmm) indicates that the job should always be active
  - Valid values for **hh** are 00 to 23.
  - Valid values for **mm** are 00 to 59.
- **command** is an optional field that you can substitute with any z/OS command

The command is executed if the targeted job is not active during the specified time interval. When ACT is specified, command cannot be blank.
**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message at five-minute intervals if the jobs in CONFIG00 are not executing:

```
dvtn 10,00
```

**Warning message**

The following warning message is issued by the DVTN sampler:

```
PWSDV8I *WARNING* asName SHOULD BE ACTIVE hhmm-hhmm
```

Message PWSDV8I indicates that the listed address space is not active during the indicated shift. You should correct the operational status of the address space.
The Enqueue Conflicts (ENQ) sampler monitors enqueue conflicts and notifies the operator when jobs have been waiting for enqueued resources 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 31.
- You can suppress reporting for specific major and minor enqueue names by including them in BBPARM member $$XENQ. See “$$XENQ” on page 39.

**Syntax**

```
ENQ 1 intervals waiting
```

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `waiting` is the number of minutes that an enqueue conflict must exist for a warning message to be issued

**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 (qqqqqqqq) 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 ENQUEUEES.

\[ n \] is the number of minutes the ENQ conflict has existed

\[ \text{Qname} \] is the Qname

\[ w \] is the number of address spaces waiting for the resource

**FIX**

The Fixed Pages (FIX) sampler monitors the amount of page-fixed common storage by its location in real memory. The page-fixed storage in the nucleus, LPA, and SQA. A warning message is issued when a threshold value is exceeded.

The monitor totals the fixed storage into one of the following groups:

**Table 5  Types of storage monitored by the FIX sampler**

<table>
<thead>
<tr>
<th>Storage type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>below-the-bar</td>
<td>storage backed by real memory with addresses less than 2G</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>backed above-the-bar</td>
<td>storage backed by real memory with addresses above 2G</td>
</tr>
</tbody>
</table>

**Syntax**

```
FIX intervals, a, b, c, d
```

The parameters are defined as follows:

\[ intervals \] is the number of report intervals between warning messages; default interval is 30 seconds

\[ a \] is the threshold for page-fixed common storage backed below-the-bar (below 2G) in kilobytes
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
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:

- `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 PWSCPMxx:

```
int 60
```

The base interval between report phases is 60 seconds.
The Job Elapsed Time (JET) sampler monitors the elapsed time of jobs and notifies the operator 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 31.

### Syntax

![Syntax Diagram]

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **class** is the job class to be monitored (A through Z and 0 through 9)
- **time** is the threshold elapsed time in minutes

**NOTE**

You can specify up to eight 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:

```
jet 4,a,5
```
**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 sssssss executing xxx minutes

where

- $n$ is the job class
- $nnnnnn$ is the JES2 job number
- $jjjjjjjj$ is the job name
- $ssssssss$ is the service class name
- $xxx$ is the elapsed time in minutes
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 31.

### Syntax

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals
  
  The default sample period is 30 seconds (one interval). You can specify up to 99.

- **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.
Examples

Assuming a report interval of 30 seconds, the following control statement displays warning messages every five minutes for each job that holds more than

- 200 slots
- 100 frames

jsu 10,100,200

Warning messages

The following warning messages are issued by the JSU sampler:

PWSJSU01 *WARNING* ttt jjjjjjjjjj 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>ttt</th>
<th>is the type, one of JOB, TSU, or STC</th>
</tr>
</thead>
<tbody>
<tr>
<td>jjjjjjjjjj</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 jjjjjjjjjj HOLDS nnnnn 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>ttt</th>
<th>is the type, one of JOB, TSU, or STC</th>
</tr>
</thead>
<tbody>
<tr>
<td>jjjjjjjjjj</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>
The Job Awaiting V=R Region (JVR) sampler monitors jobs awaiting a V=R region and notifies the operator 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 31.

**Syntax**

The parameters are defined as follows:

- `intervals`: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)

**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 jijijijj WAITING FOR V=R REGION
```

The PWSJVR01 message indicates that a job is waiting for V=R region, where

```
jijijijj
```

is the job name of the waiting job.
The Mounts Time Pending (MTP) sampler monitors tape and DASD-mount requests, and notifies the operator when outstanding mount requests have been pending for more than a specified interval.

**Syntax**

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `pending` is the number of minutes that a mount must remain pending for a warning message to be issued

**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.
The Non Reusable Queue (NRQ) sampler displays a warning message when both of the following conditions occur:

- The number of available address space vector table slots (ASIDs) that are reserved 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 drops below the specified threshold value (limit2).

**Syntax**

```
NRQ intervals limit1 limit2
```

The parameters are defined as follows:

- `intervals`: specifies the length of the sample period in intervals; the default interval is 30 seconds
- `limit1`: 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.

- `limit2`: 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.

**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 displays a warning message when both of the following conditions occur:

- The percentage of available address space vector table slots (ASIDs) 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 drops below the specified threshold value (limit2).
Syntax

```
NRQP  <intervals> <limit1> <limit2>
```

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default interval is 30 seconds
- **limit1** is the threshold for slots reserved for replacing slots marked non-reusable
  
The percentage of slots on the non-reusable replacement queue (ASVTANR / ASVTNONR * 100) must be less than or equal to the specified percentage for a warning message to be issued.
- **limit2** is the threshold for the percentage of address spaces (MAXUSER) that can still be started
  
The percentage of the original number of slots on the available queue that are still available (ASVTAAV / ASVTMAXI * 100) must be less than or equal to the specified percentage for a warning message to be issued.

**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

```
NRQP  10,25,10
```

Every 10 intervals, this NRQP sampler checks to see if less than 26% (limit1 value) of the slots remain on the non-reusable replacement queue. It also checks to see if less than 10% (limit2 value) of MAXUSER (obtained from ASVTMAXI) slots remain 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 percentage of slots on the available queue exceeds the limit2 value.

### Warning message

The following warning message is issued by the NRQP service:

```
PWSNRQP1 WHICH??? PWSNRPQ1 *WARNING* Non-reusable replacement queue length is nnn percent
```

### OUT

The Jobs Swapped Out (OUT) sampler monitors jobs, TSO sessions, and Started Tasks and reports on 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 $$INEXCL. For details, see "$$INEXCL" on page 31.

### Syntax

```
OUT [intervals] time
```

The parameters are defined as follows:

- **intervals**: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **time**: is the threshold for swap-out in minutes
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:

```
oout 2,1
```

Warning message

The following warning message is issued by the OUT sampler:

```
PSWOUT01 *WARNING* ttt jijjijjjj (nnnn) SWAPPED OUT xxxxx MINS FOR ** reason **
```

where

- **ttt** address space type: JOB, STC, or TSU
- **jjjjjjjj** job name
- **nnnn** ASID
- **xxxxx** minutes swapped out
- **reason** swap reason

The swap reason will be one of the following reasons:

- 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
The Paging Rate (PAG) sampler monitors the system paging rate and notifies the operator when the number of paging operations-per-second exceeds a specified value.

**Syntax**

```
PAG [intervals] rate
```

The parameters are defined as follows:

- `intervals`: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `rate`: is the paging rate per second measured over one reporting interval that triggers a warning message

**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.
The PCPU sampler monitors CPU utilization for the host partition relative to its relative-share value and issues 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.

**Syntax**

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default is 30 seconds (one interval).
- **lowpct** is the low threshold percent.
  
  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. A value of 50 is assumed if both **lowpct** and **highpct** are omitted.

- **highpct** is the high threshold percent.
  
  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.

- **plexpct** is the Plex minimum percent.
  
  Warning messages for the **lowpct** and **highpct** conditions are suppressed when the total Plex CPU utilization is less than the specified percentage. The default percentage is 80%. Total Plex busy includes Plex dispatching overhead.

**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

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 either 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.
The PER sampler monitors the time that the system is spending servicing an active PER SLIP trap. Because an active PER trap can cause serious performance degradation, it is essential that you be notified when a PER trap is active.

This sampler watches for excessive resource utilization due to an active trap and warns you when it sees that a trap might cause a problem.

AEW Warning message PWSPER01 is issued when a PER SLIP trap is active and its resource utilization exceeds the threshold set by the active PER interval,threshold command in the PWSCPMxx PARMLIB member.

**Syntax**

```
PER intervals threshold
```

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)

- `threshold` is the percentage of resource utilization allowed before a warning message is triggered

  A warning message is produced when this threshold is exceeded.

**Example**

Assuming a report interval of 30 seconds, the following control statement displays warning messages at two-minute intervals if the PER trap utilization exceeds 30% over the preceding two minutes:

```
PER 4,30
```
Warning message

The following warning message is issued by the PER sampler:

PWSPER01 *WARNING* PER SLIP trap xxxx active xxxxxx% of the time in the last xxxxxx seconds

This AEW Warning message is issued when a PER SLIP trap is active and its resource utilization exceeds the threshold set by the active PER interval,threshold command in the PWSCPMxx PARMLIB member.

The parameters are defined as follows:

- xxxx is the four-character SLIP trap ID specified when the SLIP was set
- xxxxxx% 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
- xxxxxx gives the time over which the samples were taken.

This time is the AEW interval in seconds times the interval field from the PER control card in your PWSCPMxx member.
The Page Data Set I/O Queue (PGQ) sampler monitors the depth of I/Os queued to page data sets and notifies the operator 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

```plaintext
PGQ intervals depth
```

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `depth` is the threshold average of I/Os queued to any page or swap data set

### 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:

PWSPGQ01 *WARNING* I/O QUEUE DEPTH FOR PAGE DATA SET nn ON vvvv IS xx

where

<table>
<thead>
<tr>
<th>nn</th>
<th>is the data set number</th>
</tr>
</thead>
<tbody>
<tr>
<td>vvvv</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>

The Replies Outstanding (REP) sampler monitors messages to the operator that require replies. This sampler notifies the operator 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 “$$RPJOBS” on page 37) or that contain specific text (see “$$RPTEXT” on page 38).

Syntax

```
REP [intervals] time
```

The parameters are defined as follows:
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:

```
PWSREP01 *WARNING* REPLIES OUTSTANDING FOR OVER nnnnnnn MINUTES
```

Message PWSREP01 indicates that replies were outstanding for nnnnnnn minutes.

**RES**

The DASD Reserves (RES) sampler monitors the reserved status of devices in a shared DASD system and notifies the operator 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 31.
- To prevent reserve information for specific volumes from displaying, add the VOLSER to BBPARM member $$XRES. See “$$XRES” on page 40.
**Syntax**

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval).
- `time` is the number of minutes that a device must be reserved for a warning message to be issued.

**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 messages**

The following warning messages are 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 and notifies the operator of 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 31.

**Syntax**

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **rptno**  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

**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

- **ttt** is STC, TSU, or JOB
- **jjjjjjjj** is the address space name
- **12345678** is the number of real-storage frames

**SMF**

The Status of SMF Recording (SMF) sampler notifies the operator if an SMF data set is full or SMF recording is not active, which ensures that no SMF data is lost.

**Syntax**

```
smf 10
```

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **threshold** is the percentage of space in all SMF data sets that has been filled

**Example**

Assuming a report interval of 30 seconds, the following control statement displays a warning message every five minutes if any SMF data set is full:

```
smf 10
```
### 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* SYS1.MAN \(n\) IS FULL; DUMP IT ASAP

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* SYS1.MAN\((n)\) IS PARTIALLY FULL AND INACTIVE; DUMP IT ASAP

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 \(xxx\) PERCENT 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.

---

**NOTE**

The SPA sampler does not support the extended area of Extended Addressability Volume (EAV) devices.

The DASD Free Space (SPA) sampler monitors free space on specified DASD volumes and notifies the operator if the free space on any specified DASD volume falls below a specified number of cylinders.
Syntax

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `cylinder` is the low threshold of free cylinders
- `volser` 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.

- `dvn` is the number of the device to be monitored; the maximum number of devices you can specify on a single statement is 14

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 FREE CYLINDER/TRACK COUNT IS xxx/yyyy
```

where

- `dvn` is the device address
- `vvvvvv` is the volume serial of device
- `xxx` is the count of free cylinders
- `yyyy` is the count of free tracks (not including cylinders)
The SPLW sampler issues a warning message for address spaces that are waiting for SPOOL space.

**Syntax**

```
SPLW 1
```

where

- `intervals` specifies the length of the report period in intervals; the default report period is 30 seconds (one interval)
- `max` 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.

**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.
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

```
1 intervals MAXUSER=nn LOWUIC=nn HIGHUIC=nn
```

The parameters are defined as follows:

- `intervals` specifies the length in intervals of the sample period; the default sample period is 30 seconds (1 interval)
- `MAXUSER=nn` is a percentage of the MAXUSER setting in SYS1.PARMLIB
- `LOWUIC=nn` is the low UIC threshold
- `HIGHUIC=nn` is the high UIC threshold

**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
```
The Swap Rate (SWP) sampler monitors system swap-out rate and notifies the operator when the number of address spaces swapped out per minute exceeds a specified value.

**Syntax**

![Diagram of SWP syntax]

The parameters are defined as follows:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `rate` is the swap rate per minute measured over one reporting interval that triggers a warning message

**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 PER MINUTE
```

Message PWSSWP01 indicates that the swap rate has exceeded the warning threshold.
The TSO CPU (TCPU) sampler generates a warning message 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 31.

**Syntax**

The parameters are defined as follows:

- **intervals**: specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **percent**: 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
- **rptno**: is the maximum number of address spaces to be listed whenever a CPU overload condition is detected; the default is 5
- **total cpu %**: is the system CPU-busy percentage that must be reached before exception messages are issued; the default is a system CPU-busy percent greater than zero

**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% FOR TSU jjjjjjj

Message PWSTCPU1 indicates that the specified address space has exceeded the warning threshold for percentage of system CPU busy time, where

- **ppp** is the address space busy percentage
- **jjjjjjjj** is the address space name

**TIO**

The TSO I/O (TIO) sampler generates a warning message 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 31.

**Syntax**

```
<table>
<thead>
<tr>
<th>TIO</th>
<th>1</th>
<th>percent</th>
<th>5</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>intervals</td>
<td>rptno</td>
<td>i/o rate</td>
<td></td>
</tr>
</tbody>
</table>
```

The parameters are defined as follows:
**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 I/O rate exceeds 50 EXCPs-per-second over the preceding two minutes:

\[ \text{tio } 4,25,5,50 \]

A warning is issued for any TSO address space doing more than 25 percent of the I/O.

**Warning messages**

The following warning messages are issued by the TIO sampler:

- **PWSTIO00**: *WARNING* TOTAL SYSTEM I/O RATE IS xxx EXCPS/SEC
  
  Message PWSTIO00 indicates that the system I/O rate has exceeded the warning threshold.

- **PWSTIO01**: *WARNING* I/O RATE FOR TSU nnnnnnn 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

  - \( nnnnnnnn \) is the address space name
  - \( xxx \) is the percent of the total system I/O rate
TPAG

The TSO Paging (TPAG) sampler sends a warning message to the operator 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 message can be suppressed if a given threshold for the total system paging rate is not reached during a reporting interval. This sampler also notifies the operator when the paging 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 31.

**Syntax**

```
TPAG 1 intervals percent rptno 25 paging rate
```

The parameters are defined as follows:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **percent** 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
- **rptno** is the maximum number of TSO address spaces to be listed whenever a TSO-paging overload condition is detected
- **paging rate** 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

**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 nnnnnnn IS xxx% OF TOTAL

Message PWSTPAG1 indicates that the specified address space has exceeded the warning threshold for percentage of system paging, where

nnnnnnnn is the TSO user ID

xxx is the percentage of the system total

TSRV

The TSO Service (TSRV) sampler sends a warning message to the operator when a TSO address space uses more than the specified percentage of one or more SRM service measures during a reporting interval. The message can be suppressed if a given total system paging 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 31.

Syntax

The parameters are defined as follows:
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.t,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 messages

The following warning messages are issued by the TSRV sampler:

```
PWSTSRV0 *WARNING* ADDR SPACE mmm SERVICE RATE IS sssss SU/SEC
```

Message PWSTSRV0 indicates that the system paging rate has exceeded the warning threshold, where

- `mmm` is the SRM service component being monitored
- `sssss` is the number of service units-per-second

```
PWSTSRV1 *WARNING* mmm SERVICE RATE FOR TSU jjjjjjjjjj IS ppp% OF SYSTEM TOTAL
```

The default is a system service rate greater than zero SUs-per-second.
Message PWSTSRV1 indicates that the specified address space has exceeded the warning threshold for percentage of system SRM component utilization, where

\[ mmm \]

is the SRM service component being monitored

\[ jjjjjjjj \]

is the address space name

\[ ppp \]

is the address space SRM component use as a percentage of system total

This Exception Monitor sampler can only be used from the MainView for z/OS product.

The Workload Delay sampler warns you 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 31.

**Syntax**

The parameters are defined as follows:

- **intervals** is the number of intervals between report periods; the default is 30 seconds (one interval)
- **workload name** is the name of the workload to be monitored
workload name* indicates a partial workload name

You can specify partial names by using a wildcard character (an asterisk).

percentage represents the minimum percentage of delay

Any delay greater than this number generates a warning message.

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.

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 wwwwwwww Delayed xxx.xx%

Message PWSWDLY0 indicates that workload wwwwwwwww has been delayed xxx.xx percent of the current MainView for z/OS data collection interval.
This Exception Monitor sampler can be used only from the MainView for z/OS product.

The Workload Objective (WOBJ) sampler warns you when a workload is not meeting its specified percentage of service objectives.

**Syntax**

```
WOBJ 1 intervals workload name* percentage
```

The parameters are defined as follows:

- `intervals` is the number of intervals between report periods; the default is 30 seconds (one interval)
- `workload name` is the name of the workload to be monitored
- `workload name*` indicates a partial workload name; you can specify partial names by using a wildcard character (an asterisk)
- `percentage` 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.

**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:

```
PWSWOBJO *WARNING* Workload wwwwwwwww at xxx% of its objective
```
Message PWSOBJ0 indicates that workload \textit{wxxxxxxxx} has met \textit{xxx} percent of its objective.

\textbf{WRT}

This Exception Monitor sampler can be used \textit{only} from the MainView for z/OS product.

The Workload Response Time (WRT) sampler warns you when the response time for workloads that match your specified selection criteria reaches or exceeds the threshold you specified.

\textbf{Syntax}

The parameters are defined as follows:

- \textit{intervals} is the length of the report period in intervals; the default is 30 seconds (one interval)
- \textit{workload name} is the name of the workload to be monitored
- \textit{workload 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.
- \textit{period} represents the performance period for the workload, which is to be used when checking the response time against the threshold value
- \textit{seconds} is the response-time threshold expressed in hundredths of a second
## Example

Assuming a report interval of 30 seconds, this example illustrates the use of four separate sampler definitions:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wrt 2</td>
<td>tso*</td>
<td>1</td>
<td>70</td>
</tr>
<tr>
<td>wrt 2</td>
<td>tso*</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>wrt 2</td>
<td>tso*</td>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>wrt 2</td>
<td>tso*</td>
<td>4</td>
<td>400</td>
</tr>
</tbody>
</table>

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:

- `intervals` specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- `threshold` is the percentage of console buffers in use that causes a warning message to be issued; the default threshold is 75 percent

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:

```
wt0 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:

- **intervals** specifies the length of the sample period in intervals; the default sample period is 30 seconds (one interval)
- **threshold** is the percentage of console buffers in use that causes a warning message to be issued; the default threshold is 75 percent

## 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:

```
wtor 1,80
```

## Warning message

The following warning message is issued by the WTOR sampler:

```
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 issues a warning message 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:

- **intervals**: specifies the length of the sample period in intervals; the default is 30 seconds (one interval)
- **lowpct**: is the low threshold percent
  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 warning message is issued if the total Plex CPU utilization during the period 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.

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 is less than 90% utilized during the period.
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.
The services offered by MainView SYSPROG Services help improve and increase productivity by allowing you to manipulate various z/OS internals. This chapter provides detailed information about services, including proper syntax, required and optional parameters, and usage examples. For information about using the Services Menu, see Chapter 3, “Using MainView SYSPROG Services.”

This chapter includes the following topics; the services marked with an asterisk are distributed with and available from the MainView AutoOPERATOR product. The remaining services are available only if you have a MainView SYSPROG Services license.

<table>
<thead>
<tr>
<th>Service name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEW</td>
<td>151</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>152</td>
</tr>
<tr>
<td>APF</td>
<td>153</td>
</tr>
<tr>
<td>ASM *</td>
<td>157</td>
</tr>
<tr>
<td>ASVT *</td>
<td>164</td>
</tr>
<tr>
<td>AUTHTSO</td>
<td>165</td>
</tr>
<tr>
<td>BBXS *</td>
<td>169</td>
</tr>
<tr>
<td>CDE</td>
<td>171</td>
</tr>
<tr>
<td>CLEAR</td>
<td>174</td>
</tr>
<tr>
<td>CNSL</td>
<td>176</td>
</tr>
<tr>
<td>COMMAND</td>
<td>178</td>
</tr>
<tr>
<td>CONSOLES</td>
<td>179</td>
</tr>
<tr>
<td>CPU *</td>
<td>180</td>
</tr>
<tr>
<td>CSA *</td>
<td>182</td>
</tr>
<tr>
<td>CSMON</td>
<td>185</td>
</tr>
<tr>
<td>CTCB</td>
<td>185</td>
</tr>
<tr>
<td>DEVIATN</td>
<td>188</td>
</tr>
<tr>
<td>DFLIST</td>
<td>190</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTORAGE *</td>
<td>346</td>
</tr>
<tr>
<td>QUEUE</td>
<td>352</td>
</tr>
<tr>
<td>DONTSWAP</td>
<td>192</td>
</tr>
<tr>
<td>DSNAME</td>
<td>193</td>
</tr>
<tr>
<td>DSSUM</td>
<td>195</td>
</tr>
<tr>
<td>DUMP</td>
<td>198</td>
</tr>
<tr>
<td>DVIEW</td>
<td>203</td>
</tr>
<tr>
<td>EDTINFO</td>
<td>209</td>
</tr>
<tr>
<td>ENCLACVE</td>
<td>215</td>
</tr>
<tr>
<td>ENQUEUES *</td>
<td>217</td>
</tr>
<tr>
<td>EQUATE *</td>
<td>221</td>
</tr>
<tr>
<td>ESCASS *</td>
<td>223</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>228</td>
</tr>
<tr>
<td>EXIT</td>
<td>231</td>
</tr>
<tr>
<td>FCOMMON *</td>
<td>232</td>
</tr>
<tr>
<td>FINDMBR *</td>
<td>233</td>
</tr>
<tr>
<td>HELP</td>
<td>236</td>
</tr>
<tr>
<td>IFA *</td>
<td>239</td>
</tr>
<tr>
<td>Service name</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>IPLDATA *</td>
<td>245</td>
</tr>
<tr>
<td>ISMF</td>
<td>249</td>
</tr>
<tr>
<td>JCPU</td>
<td>251</td>
</tr>
<tr>
<td>JITOKEN *</td>
<td>253</td>
</tr>
<tr>
<td>JSTORM</td>
<td>254</td>
</tr>
<tr>
<td>LABEL *</td>
<td>256</td>
</tr>
<tr>
<td>LCPU *</td>
<td>257</td>
</tr>
<tr>
<td>LNKLIST</td>
<td>259</td>
</tr>
<tr>
<td>LOADLPA</td>
<td>266</td>
</tr>
<tr>
<td>LOGGING *</td>
<td>269</td>
</tr>
<tr>
<td>LWAIT *</td>
<td>272</td>
</tr>
<tr>
<td>LX *</td>
<td>275</td>
</tr>
<tr>
<td>MCOMMAND</td>
<td>279</td>
</tr>
<tr>
<td>MDEVICE *</td>
<td>280</td>
</tr>
<tr>
<td>MEMSCAN *</td>
<td>284</td>
</tr>
<tr>
<td>MIO *</td>
<td>290</td>
</tr>
<tr>
<td>MLCU *</td>
<td>293</td>
</tr>
<tr>
<td>MONITOR *</td>
<td>295</td>
</tr>
<tr>
<td>MPATH *</td>
<td>297</td>
</tr>
<tr>
<td>MSTORAGE *</td>
<td>299</td>
</tr>
<tr>
<td>MTP *</td>
<td>301</td>
</tr>
<tr>
<td>NTB</td>
<td>302</td>
</tr>
<tr>
<td>OFIND *</td>
<td>306</td>
</tr>
<tr>
<td>OKSWAP</td>
<td>310</td>
</tr>
<tr>
<td>OLIST *</td>
<td>312</td>
</tr>
<tr>
<td>PAGING *</td>
<td>317</td>
</tr>
<tr>
<td>PARMLIST</td>
<td>318</td>
</tr>
<tr>
<td>PBUILD</td>
<td>320</td>
</tr>
<tr>
<td>PFKEY</td>
<td>321</td>
</tr>
<tr>
<td>PIO</td>
<td>323</td>
</tr>
<tr>
<td>PLIST</td>
<td>324</td>
</tr>
<tr>
<td>POST</td>
<td>325</td>
</tr>
<tr>
<td>PPT *</td>
<td>326</td>
</tr>
<tr>
<td>PRIVATE *</td>
<td>331</td>
</tr>
<tr>
<td>PROGRESS *</td>
<td>337</td>
</tr>
<tr>
<td>PRSM</td>
<td>342</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFO *</td>
<td>240</td>
</tr>
<tr>
<td>IO *</td>
<td>242</td>
</tr>
<tr>
<td>REPLIES *</td>
<td>360</td>
</tr>
<tr>
<td>RESERVES *</td>
<td>362</td>
</tr>
<tr>
<td>RSM *</td>
<td>364</td>
</tr>
<tr>
<td>SCT</td>
<td>369</td>
</tr>
<tr>
<td>SDUMP</td>
<td>370</td>
</tr>
<tr>
<td>SETOP *</td>
<td>374</td>
</tr>
<tr>
<td>SMFEXIT</td>
<td>376</td>
</tr>
<tr>
<td>SMSLIST</td>
<td>381</td>
</tr>
<tr>
<td>SOFTFRR *</td>
<td>385</td>
</tr>
<tr>
<td>SP2</td>
<td>390</td>
</tr>
<tr>
<td>SPACE *</td>
<td>390</td>
</tr>
<tr>
<td>SRM *</td>
<td>398</td>
</tr>
<tr>
<td>SSVT</td>
<td>401</td>
</tr>
<tr>
<td>STATUS *</td>
<td>403</td>
</tr>
<tr>
<td>SUBMIT</td>
<td>405</td>
</tr>
<tr>
<td>SVCFIN</td>
<td>407</td>
</tr>
<tr>
<td>SYSDUMP *</td>
<td>410</td>
</tr>
<tr>
<td>TCB</td>
<td>411</td>
</tr>
<tr>
<td>TIME</td>
<td>415</td>
</tr>
<tr>
<td>TIOT *</td>
<td>417</td>
</tr>
<tr>
<td>TOD</td>
<td>420</td>
</tr>
<tr>
<td>TPIO *</td>
<td>422</td>
</tr>
<tr>
<td>TQE *</td>
<td>425</td>
</tr>
<tr>
<td>TRACK *</td>
<td>427</td>
</tr>
<tr>
<td>TSULIST *</td>
<td>431</td>
</tr>
<tr>
<td>UCB *</td>
<td>433</td>
</tr>
<tr>
<td>USING *</td>
<td>435</td>
</tr>
<tr>
<td>VIO</td>
<td>437</td>
</tr>
<tr>
<td>VMCMD *</td>
<td>438</td>
</tr>
<tr>
<td>WARNING</td>
<td>439</td>
</tr>
<tr>
<td>WHERE *</td>
<td>442</td>
</tr>
<tr>
<td>ZAP</td>
<td>444</td>
</tr>
<tr>
<td>ZIIP *</td>
<td>446</td>
</tr>
<tr>
<td>VIO</td>
<td>437</td>
</tr>
</tbody>
</table>
Alias for WARNING. See the service “WARNING” on page 439.
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:

```
AMTAN51 3 Address spaces are waiting to be dispatched
```

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

```
AMTAN11 2 waiting for resource held by VAM3NQ1 (0061) SJSO?AMTAN21QNAME=PINION,RNAME=Test_ENQ_Rname
```
This service

- lists the current authorized program libraries and their volume serial numbers
- 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**

```plaintext
APF
    - dsname
    - partialDsname
    - volser
    - dsname
    - volser
    +SMS*
    +*****

ADD
    - volser
    +SMS*
    +*****

DELETE
    - volser
    +SMS*
    +*****

CHANGE
    - volser
    +SMS*
    +*****
```
The parameters are defined as follows:

- **dsname**: data set name of the authorized library
- **partialDsname**: represents the first characters of a data set name
- **volser**: volume serial number

For instance, if you type **APF RXA15**, the service lists every data set that has a prefix of RXA15.

**Legend:**
1. hexadecimal address of the APF list in storage
2. APF list

**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`

```
AMTA77I       THE FOLLOWING ENTRIES MATCHED THE REQUEST
AMTA72I       VOLUME OPNS00   DSN RES1.RES31.BBLINK
```

---

**Example 3**

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

`apf,opns00`

```
AMTA72I       VOLUME OPNS00   DSN RES1.RES30.BBLINK
```

Note the two commas, indicating 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`

```
AMTA73I       APF LIST NOW CONTAINS ENTRY FOR
AMTA72I       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`
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
```

**Example 7**

To remove an entry from the APF list, type

```
apf res1.res30.bbload,opns01,delete
```

**Example 8**

To change the volser of an existing APF entry, type

```
apf res1.res30.bbload,opns01,change,perf01
```
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

The parameters are defined as follows:

- **STATUS** displays information and unusual ASM-related conditions
  
  STATUS 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.

- **DSN** displays information and unusual ASM-related conditions
  
  DSN 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.

When any of the following parameters are specified, a table containing the following information is displayed for each space-holding slot sorted by that parameter: jobname, ASID, total auxiliary storage slots in use by the address space, the number of non-VIO slots in use, the number of VIO slots in use, and the percentage of the slots in use to the total number of slots in the local page data sets.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASID</td>
<td>displays information sorted in ascending ASID sequence</td>
</tr>
<tr>
<td></td>
<td>All address spaces are displayed unless limited to ( nnn ) by the second parameter.</td>
</tr>
<tr>
<td>MAP</td>
<td>is an alias for ASID to maintain compatibility with prior releases</td>
</tr>
<tr>
<td>NAME</td>
<td>sorts information in ascending jobname sequence</td>
</tr>
<tr>
<td></td>
<td>All address spaces are displayed unless limited to ( nnn ) by the second parameter.</td>
</tr>
<tr>
<td>TOTAL</td>
<td>sorts information in descending TOTAL sequence (sum of non-VIO and VIO slots in use by the address space)</td>
</tr>
<tr>
<td></td>
<td>The display is limited to the top ten address spaces unless ( nnn ) is specified as the second parameter, in which case the top ( nnn ) address spaces are displayed.</td>
</tr>
<tr>
<td>NVIO</td>
<td>sorts information in descending NVIO sequence (number of non-VIO slots in use by the address space)</td>
</tr>
<tr>
<td></td>
<td>The display is limited to the top ten address spaces unless ( nnn ) is specified as the second parameter, in which case the top ( nnn ) address spaces are displayed.</td>
</tr>
<tr>
<td>VIO</td>
<td>sorts information in descending VIO sequence (number of VIO slots in use by the address space)</td>
</tr>
<tr>
<td></td>
<td>The display is limited to the top ten address spaces unless ( nnn ) is specified as the second parameter, in which case the top ( nnn ) address spaces are displayed.</td>
</tr>
<tr>
<td>( nnn )</td>
<td>limits the display to the first ( nnn ) address spaces</td>
</tr>
<tr>
<td></td>
<td>When specified in the first parameter position, NVIO is implied.</td>
</tr>
<tr>
<td></td>
<td>The default is 10 for displays sorted by TOTAL, NVIO, or VIO. The default is ALL for displays sorted by ASID or NAME.</td>
</tr>
<tr>
<td>ALL</td>
<td>indicates that the information for all address spaces is to be displayed</td>
</tr>
<tr>
<td></td>
<td>This parameter is the default when ASID or NAME is specified.</td>
</tr>
</tbody>
</table>
**Column headers**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NUM</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>DVN</strong></td>
<td>The four-character hexadecimal device number.</td>
</tr>
<tr>
<td><strong>DSN</strong></td>
<td>The data set name.</td>
</tr>
<tr>
<td><strong>VOLSER</strong></td>
<td>The volume serial number of the volume containing the page data set.</td>
</tr>
<tr>
<td><strong>#Ex</strong></td>
<td>The number of extents allocated for the page data set.</td>
</tr>
<tr>
<td><strong>E</strong></td>
<td>VIO eligibility status. ( Y ) indicates the data set is eligible for VIO use and ( N ) indicates it is not eligible for VIO use.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of paging data set. <strong>PLPA</strong> for PLPA, <strong>Comm</strong> for Common, and <strong>Local</strong> for Local pages data sets.</td>
</tr>
<tr>
<td><strong>Slots</strong></td>
<td>The total number of allocated slots (pages) available in the data set.</td>
</tr>
<tr>
<td><strong>Pct Free</strong></td>
<td>The percentage of the allocated slots that are currently available for use.</td>
</tr>
<tr>
<td><strong>Jobname</strong></td>
<td>The name of the job currently executing in the address space.</td>
</tr>
<tr>
<td><strong>ASID</strong></td>
<td>The Address Space Identifier (ASID) for the address space.</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Non-VIO</strong></td>
<td>The number of non-VIO local page data set slots in use by address space.</td>
</tr>
<tr>
<td><strong>VIO</strong></td>
<td>The number of VIO local page data set slots in use by the address space.</td>
</tr>
<tr>
<td><strong>Pct Used</strong></td>
<td>The percentage of slots used by this address space to the total number of allocated local page data set slots.</td>
</tr>
</tbody>
</table>

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

Note the warning message (AMTA59W). This message and other warning messages are displayed when an unusual condition is present.

```
AMTA5CI Last IPL was Cold Start (CLPA)
AMTA52I Total local slots= 4,320K Available= 3,633K ( 84%)
AMTA5DI Largest holder of slots is TRACE with 1.13%
AMTA5GI
AMTA59W PLPA data set spilled into common
AMTA5GI
AMTA5HI NUM  DVN  VOLSER  #Ex  E  Type     Slots   Pct Free
AMTA5II ---  ----  ------  ---  -  -----  --------  --------
AMTA53I   0  85B3  SADPG1  1  Y  PLPA    17,999     0.00
AMTA53I   1  85B3  SADPG1  1  Y  Comm    53,999    93.98
AMTA53I   2  Not in use
AMTA53I   3  85B3  SADPG1  1  Y  Local    135K    79.94
AMTA53I   4  85B3  SADPG1  1  Y  Local    135K    73.69
AMTA53I   5  85B3  SADPG1  1  Y  Local    53,999   21.79
AMTA53I   6  A205  SADPG2  1  Y  Local    180K    83.79
AMTA53I   7  A205  SADPG2  1  Y  Local    180K    81.45
AMTA53I   8  A205  SADPG2  1  Y  Local    180K    79.64
AMTA53I   9  4A10  SADPG3  1  Y  Local    180K    73.73
AMTA53I  10  4A10  SADPG3  1  Y  Local    180K    73.63
AMTA53I  11  4A10  SADPG3  1  Y  Local    180K    74.36
AMTA53I  12  4A1F  SADPG4  1  Y  Local    180K    73.82
AMTA53I  13  4A1F  SADPG4  1  Y  Local    180K    73.28
AMTA53I  14  4A1F  SADPG4  1  Y  Local    180K    72.77
AMTA53I  15  495A  SADPG5  1  Y  Local    594K    91.91
AMTA53I  16  85E5  SADPG6  1  Y  Local    594K    91.21
AMTA53I  17  A202  SADPG7  1  Y  Local    594K    91.90
AMTA53I  18  495E  SADPG8  1  Y  Local    594K    91.69
AMTA53I  19  Not in use
AMTA53I  20  Not in use
AMTA53I  21  Not in use
AMTA53I  22  Not in use
AMTA53I  23  Not in use
AMTA53I  24  Not in use
```

160  *MainView SYSPROG Services User Guide and Reference*
### 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

\texttt{asm,dn}

<table>
<thead>
<tr>
<th>AMTA5CI</th>
<th>Last IPL was Cold Start (CLPA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTA52I</td>
<td>Total local slots = 4,320K Available = 3,633K (84%)</td>
</tr>
<tr>
<td>AMTA5DI</td>
<td>Largest holder of slots is TRACE with 1.13%</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>PLPA data set spilled into common</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTA5HI</th>
<th>NUM</th>
<th>Data Set Name</th>
<th>Slots</th>
<th>Pct Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTA50I</td>
<td>0</td>
<td>PAGE.SJSD.PLPA</td>
<td>17,999</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>1</td>
<td>PAGE.SJSD.COMMON</td>
<td>53,999</td>
<td>93.98</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>2</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>3</td>
<td>PAGE.SJSD.LOCAL1</td>
<td>135K</td>
<td>79.94</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>4</td>
<td>PAGE.SJSD.LOCAL2</td>
<td>135K</td>
<td>73.69</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>5</td>
<td>PAGE.SJSD.LOCAL3</td>
<td>53,999</td>
<td>21.79</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>6</td>
<td>PAGE.SJSD.LOCAL4</td>
<td>180K</td>
<td>83.79</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>7</td>
<td>PAGE.SJSD.LOCAL5</td>
<td>180K</td>
<td>81.45</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>8</td>
<td>PAGE.SJSD.LOCAL6</td>
<td>180K</td>
<td>79.64</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>9</td>
<td>PAGE.SJSD.LOCAL7</td>
<td>180K</td>
<td>73.73</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>10</td>
<td>PAGE.SJSD.LOCAL8</td>
<td>180K</td>
<td>73.63</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>11</td>
<td>PAGE.SJSD.LOCAL9</td>
<td>180K</td>
<td>74.36</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>12</td>
<td>PAGE.SJSD.LOCALA</td>
<td>180K</td>
<td>73.82</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>13</td>
<td>PAGE.SJSD.LOCALB</td>
<td>180K</td>
<td>73.28</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>14</td>
<td>PAGE.SJSD.LOCALC</td>
<td>180K</td>
<td>72.77</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>15</td>
<td>PAGE.SJSD.LOCALD</td>
<td>594K</td>
<td>91.91</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>16</td>
<td>PAGE.SJSD.LOCALE</td>
<td>594K</td>
<td>91.21</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>17</td>
<td>PAGE.SJSD.LOCALF</td>
<td>594K</td>
<td>91.90</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>18</td>
<td>PAGE.SJSD.LOCALG</td>
<td>594K</td>
<td>91.69</td>
</tr>
<tr>
<td>AMTA50I</td>
<td>19</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>20</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>21</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>22</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>23</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTA50I</td>
<td>24</td>
<td>Not in use</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AMT001A SYSPROG
Example 3

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

`asm, total`

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

```
AMTA55I There are no address spaces holding local slots
AMT001A SYSPROG
```
### 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>AMTA5JI</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>AMTA5KI</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>WS70BSS</td>
<td>0111</td>
<td>91,574</td>
<td>91,574</td>
<td>0 2.11</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>WS61BSS</td>
<td>00D8</td>
<td>87,954</td>
<td>87,954</td>
<td>0 2.03</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>WS70BSA</td>
<td>0110</td>
<td>80,210</td>
<td>80,210</td>
<td>0 1.85</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>WS70BS</td>
<td>00F7</td>
<td>67,600</td>
<td>67,600</td>
<td>0 1.56</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>OMVS</td>
<td>000F</td>
<td>62,599</td>
<td>62,599</td>
<td>0 1.44</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>WS61BS</td>
<td>00FF</td>
<td>59,014</td>
<td>59,014</td>
<td>0 1.36</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>TRACE</td>
<td>0004</td>
<td>49,341</td>
<td>49,341</td>
<td>0 1.14</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>EYUX410</td>
<td>00B4</td>
<td>33,931</td>
<td>33,931</td>
<td>0 0.78</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DIB1DBM1</td>
<td>007E</td>
<td>30,730</td>
<td>30,730</td>
<td>0 0.71</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTQ7MSTR</td>
<td>00AF</td>
<td>28,342</td>
<td>28,342</td>
<td>0 0.65</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DOH02DBM1</td>
<td>00A7</td>
<td>27,290</td>
<td>27,290</td>
<td>0 0.63</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DHH1DBM1</td>
<td>0089</td>
<td>26,619</td>
<td>26,619</td>
<td>0 0.61</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DEE0DBM1</td>
<td>0088</td>
<td>26,542</td>
<td>26,542</td>
<td>0 0.61</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DJB1DBM1</td>
<td>018C</td>
<td>25,957</td>
<td>25,957</td>
<td>0 0.60</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>DID2DBM1</td>
<td>0192</td>
<td>25,952</td>
<td>25,952</td>
<td>0 0.60</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>RASP</td>
<td>0003</td>
<td>24,148</td>
<td>24,148</td>
<td>0 0.55</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>ZFS</td>
<td>003F</td>
<td>22,489</td>
<td>22,489</td>
<td>0 0.52</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTA6MSTR</td>
<td>00AC</td>
<td>22,487</td>
<td>22,487</td>
<td>0 0.52</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>CSQ7MSTR</td>
<td>0187</td>
<td>20,497</td>
<td>20,497</td>
<td>0 0.47</td>
</tr>
<tr>
<td>AMTA54I</td>
<td>MTAPASD</td>
<td>00BA</td>
<td>15,858</td>
<td>15,858</td>
<td>0 0.36</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example 5

To display to the first four address spaces, type

```
asm,4
```

Specifying `asm,4` is functionally the same as specifying `ASM,NVIO,4`. 
ASVT

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

- 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

```
ASVT
```

Example

To display information for the three address space queues, type

```
asvt
```

<table>
<thead>
<tr>
<th>Description</th>
<th>Current</th>
<th>Percent</th>
<th>Original</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABLE QUEUE</td>
<td>225</td>
<td>45.00</td>
<td>500</td>
<td>MAXUSER</td>
</tr>
<tr>
<td>NON-REUSABLE REPLACEMENT QUEUE</td>
<td>43</td>
<td>28.66</td>
<td>150</td>
<td>RSVNONR</td>
</tr>
<tr>
<td>START/SASI QUEUE</td>
<td>25</td>
<td>100.00</td>
<td>25</td>
<td>RSVSTRT</td>
</tr>
<tr>
<td>NUMBER OF NON-REUSABLE ASID</td>
<td>107</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The AUTHTSO (AUT) service

- lists the contents of the current authorization lists
- builds a new authorization list from any IKJTSOxx 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**
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 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>
<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>PGM</td>
<td>specifies authorized TSO programs</td>
</tr>
<tr>
<td>TSF</td>
<td>specifies the TSO Service Facility</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

`authhtso list cmd`

```
[1] AMTAT7I AUTHORIZED COMMAND LIST (IKJEFTE2) -  
[2] [3] AMTAT7I BUILT DURING IPL AT 02:16 ON 96.015:  
AMTAT7I RECEIVE TRANSMIT XMIT LISTD [4]  
AMTAT8I SEND RACONVRT SYNC SYSPROG [4]  
```

**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 (IKJEFT2E) -
AMTAT7I BUILT DURING IPL AT 02:16 ON 96.022:
AMTAT8I RECEIVE TRANSMIT XMIT LISTD
AMTAT8I SEND RACONVRT SYNC SYSPROG
AMTAT7I
AMTAT7I AUTHORIZED PROGRAM LIST (IKJEFT2E) -
AMTAT7I BUILT DURING IPL AT 02:16 ON 96.022:
AMTAT8I IEBCOPY SPFCOPY IKJEFF76 SYSPROG
AMTAT8I ICHUT100 ICHUT200 ICHUT400
AMTAT7I
AMTAT7I AUTHORIZED APF-TSF LIST (IKJEFTAP) -
AMTAT7I BUILT DURING IPL AT 02:16 ON 96.022:
AMTAT7I
```

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
y
AMTATBI TSO CMD LIST (IKJEFT2E) 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
y
AMTATBI TSO CMD LIST (IKJEFT2E) RESTORED
AMTAT9A REPLY Y TO CONFIRM RESTORE OF TSO PGM LIST, N TO CANCEL
y
AMTATBI TSO PGM LIST (IKJEFT2E) RESTORED
AMTAT9A REPLY Y TO CONFIRM RESTORE OF TSO TSF LIST, N TO CANCEL
y
AMTATBI 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.
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:

- **BBXS** displays information about the BBXS subsystem (default)
- **subsystemID** displays information about the specified BMC Software subsystem service (when it is named something other than BBXS)
- **ALL** displays a summary line for each BBXS subsystem ID

**Example 1**

To display information about the BBXS subsystem, type

```
bbxs
```

---

```
<table>
<thead>
<tr>
<th>AMTBXAD</th>
<th>BBX Subsystem Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTBXDD</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>AMTBX2D</td>
<td>Subsystem name=BBXS; SSCT=00BD3270 ## AUTOINIT ##</td>
</tr>
<tr>
<td>AMTBX3D</td>
<td>BBCT addr=00B5A000; restart count=01; RMID=BPB1160 , LEVEL=0005</td>
</tr>
<tr>
<td>AMTBX5D</td>
<td>Initialized by ITSTNPAS (mod=CX10SDVR.key=4) at 03:49:32 on 12/17/95</td>
</tr>
<tr>
<td>AMTBX6D</td>
<td>BBX was loaded from BB.ITSTN.BBLINK</td>
</tr>
</tbody>
</table>

| AMTBX4D | Previous BBCT Addr=00B9F000; RMID=BPB1144 , level=0005 ## AUTOINIT ## |
| AMTBX5D | Initialized by DC$BBI   (mod=ASTXA1MN.key=8) at 03:47:48 on 12/17/95 |
| AMTBX6D | BBX was loaded from SYSO.$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
```

<table>
<thead>
<tr>
<th>AMTBXAS All Subsystems Associated with BBX</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTBXAA -----------------------------------</td>
</tr>
<tr>
<td>AMTBX9D Subsystem name=BBXS; SSCT=00BD8008, BBCT=00BD000</td>
</tr>
</tbody>
</table>
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**

The parameters are defined as follows:

- `@` specifies your own address space
- `asi` address space instance; see “Identifying address spaces” on page 17 for more information
- `modname` limits the display to the specified module
- `D` displays the three-attribute bytes and the address of the CDE in hexadecimal instead of mnemonics for selected attributes

**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

<table>
<thead>
<tr>
<th>AMTY401</th>
<th>Address Space: PCAUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTY411</td>
<td></td>
</tr>
<tr>
<td>AMTY421</td>
<td>TCB address is 5FE1BB</td>
</tr>
<tr>
<td>AMTY431</td>
<td>Listing of associated CDE's</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A Load Adr Length Entry Pt Loc Usect Attributes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEAVAR00</td>
<td>081D31C8</td>
<td>23F8</td>
<td>881D31C8</td>
<td>EFLPA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTY451</th>
<th>AMTY461</th>
<th>AMTY411</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Space: PCAUTH</td>
<td>TCB address is 5FDE2B</td>
<td></td>
</tr>
</tbody>
</table>
Example 2 (with D)

cde,pcauth,d

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)
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 370 and “SYSDUMP” on page 410 for more information.

**NOTE**

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

### Syntax

```
CLEAR  ALL        PROMPT
       id         NOPROMPT
```

The parameters are defined as follows:

- **ALL** clears all system dump data sets (default)
- **id** displays a one- or two-digit number representing the system dump data set to be cleared
- **PROMPT** lists the title of the dump contained in each full dump data set processed; in addition, the system issues a confirmation message

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
```

<table>
<thead>
<tr>
<th>AMTC50I</th>
<th>SYS1.DUMP01 WAS FILLED AT 10:20 ON JAN 21,2003 [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC51I</td>
<td>SOURCE=SVCDUMP [2]</td>
</tr>
<tr>
<td>AMTC52I</td>
<td>TITLE=LOW CORE OVERLAY (LOOP) [3]</td>
</tr>
<tr>
<td>AMTC50A</td>
<td>ENTER Y TO CLEAR SYS1.DUMP01 [4]</td>
</tr>
<tr>
<td></td>
<td>y</td>
</tr>
<tr>
<td>AMTC53I</td>
<td>'SYS1.DUMP00' DATA SET CLEARED [5]</td>
</tr>
<tr>
<td>AMTC5AI</td>
<td>CLEAR      PROCESSING COMPLETE</td>
</tr>
</tbody>
</table>

**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—ALL
  N—NOPROMPT
  Y—PROMPT
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.

**Syntax**

```plaintext
CNSL
```

**Example**

To simulate an operator console, type

```
cnsl
```

```
*14.10.37 SJSC STC05665 *.MIM9001 W MIMGR cannot validate GDIF product
*license until CAIRIM services are available
*14.10.37 SJSC STC05672 *.MIM9001 W MIMCON cannot validate GCMF product
*license until CAIRIM services are available
*14.10.37 SJSC STC05672 *.MIM9001 W MIMCON cannot validate ICMF product
*license until CAIRIM services are available
*14.13.09 SJSC STC08258 *TX3006I Forcing IPL mode
-14.14.49 SJSC STC07218 +NEARING JOB WAIT LIMIT
-14.14.54 SJSC JOB07072 +ITP137I MVQPRD42 T3 -00001-MVQ03 is
-monitoring from CONACTZ #075.
-14.14.58 SJSC JOB06831 +ITP137I MVQPRD50 T53 -00001-MVQ53 is
-monitoring from PLEX #083.
14.15.19 SJSC STC07056 .MIM1038 CICS6MVI contention with CICS6MVI owns
EXCL on SJSC
14.15.19 SJSC STC07056 .MIM1039 CICS6MVI needs EXCL SYSDSN
MVIMS.CICS620.DFHLRQ
*14.15.19 SJSC STC07056 *.MIM1040 CICS6MVI waiting for resources for 264.
*minutes
00- 14.15.29 SJSC JOB08008 DFS27311 MSDB checkerpoint started I8A
-14.15.29 SJSC JOB08008 DFS9941*CHKPT 04292/141529**SIMPLE* I8A

AMTCN4I CONSOLE-3020 RTME=02,RNUM=20,SEG=20,CON=N, DEL=RD
AMTCN5I MODE=RESOLVE ACTION MESSAGES=5,REPLIES=0,INT REQ=0
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
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.
The COMMAND (COM) service provides the ability to execute any system 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:

- `command` operator command

**Example 1**

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

```
com,set dae=01
```

Output:

```
AMTC221 ADY015I DAE STOP PROCESSING IS COMPLETE
AMT001A SYSPROG
```

**Example 2**

To execute the command D U,TAPE, type

```
com,d u,tape
```

Output:

```
AMTC22I IEE457I 06.38.19 UNIT STATUS 250
AMTC22I 01A0 3490 OFFLINE /REMOV
AMTC22I 01A1 3490 OFFLINE /REMOV
AMTC22I 01A2 3490 OFFLINE /REMOV
AMTC22I 01A3 3490 OFFLINE /REMOV
AMTC22I 01A4 3490 OFFLINE /REMOV
AMTC22I 01A5 3490 OFFLINE /REMOV
AMTC22I 01A6 3490 OFFLINE /REMOV
AMTC22I 01A7 3490 OFFLINE /REMOV
AMTC22I 01A8 3490 OFFLINE /REMOV
AMTC22I 01A9 3490 OFFLINE /REMOV
```
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
```

```
[1] CONSOLE 0A0 HAS 1 MESSAGES TO BE DISPLAYED
[2] CONSOLE 0A2 HAS 198 MESSAGES TO BE DISPLAYED
[3] 198 OF 200 CONSOLE BUFFERS IN USE
[4] 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 by the system

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**

```
CPU [time] 10
```

The parameter is defined as follows:

| time          | length of the sample period in seconds; the default is 10 seconds, and the maximum value is 1,200 seconds |
Example

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

cpu, 10

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

4. percentage of the sample period that the partition hosting the SYSROG 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.
5. 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.

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

7. 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.

8. CPU usage by address space

9. priority of the address space

10. CPU usage for all other address spaces in the LPAR

11. 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)

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

**CSA**

The CSA service displays the utilization of

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

**Syntax**

```
CSA SUM
| MAP
| BOTH
```
The parameters are defined as follows:

**SUM** provides a summary of common storage (CSA and SQA) usage
**MAP** provides a detailed display of CSA/ECSA usage by subpool and storage key
**BOTH** provides both the summary and the detailed displays

### Example 1

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

`csa,map`

<table>
<thead>
<tr>
<th>AMTCMAI Owner</th>
<th>SUP</th>
<th>SCHED</th>
<th>VSPC</th>
<th>AVM</th>
<th>DM</th>
<th>VTAM</th>
<th>IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTCMBI SP/Key</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>AMTCMCI Non-Extended CSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI 227 17Ki 0 0 0 11Ki 0 24Ki 21Ki 73Ki</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI 228 233Ki 0 0 0 43Ki 0 48Ki 79Ki 406Ki</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI 231 1Ki 0 0 0 0 0 20Ki 460Ki 481Ki</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMDI 241 2Mi 34Ki 35Ki 32 219Ki 4Ki 2Mi 4Mi 5Mi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTCMEI Total 0 2Mi 37Ki 35Ki 32 273Ki 4Ki 99Ki 2Mi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| AMTCMDI Extended CSA | | | | | | | | | |
| AMTCMDI 227 150Ki 147Ki 0 144 198Ki 4Ki 4Mi 1Mi 5Mi |
| AMTCMDI 228 10Mi 13Ki 215Ki 0 44Mi 2Mi 2Mi 10Mi 67Mi |
| AMTCMDI 231 2Mi 11Ki 0 13Ki 979Ki 2Mi 6Mi 48Mi 59Mi |
| AMTCMDI 241 87Mi 1Mi 2Mi 58Ki 25Mi 1Mi 11Mi 45Mi 173Mi |
| AMTCMEI Total 0 99Mi 1Mi 2Mi 72Ki 70Mi 5Mi 22Mi 200Mi |

**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.**

   - 0 supervisor and other system functions that require access to all areas of storage
   - 1 job scheduler, job entry subsystem (JES), APC, and TSO/E
   - 2 (reserved)
   - 3 availability manager (AVM)
   - 4 (reserved)
2. one line per subpool

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

| AMTCM0I |  |  |
|---------|  |  |
| AMTCM0I | Size | 3,816K | 201M [1] |
| AMTCM1I | Converted to SQA | 0K | 0.0% | 0M | 0.0% [2] |
| AMTCM2I | Allocated | 814K | 21.3% | 36M | 18.1% [3] |
| AMTCM3I | High-water mark | 822K | 21.6% | 37M | 18.4% [4] |
| AMTCM4I | Available | 3,002K | 78.7% | 165M | 81.9% [5] |
| AMTCM5I | Largest avail blk | 2,908K | 76.2% | 161M | 80.3% |

| AMTCM5I |  |  |
|---------|  |  |
| AMTCM5I | Size (Incl Conv) | 1,816K | 18M [1] |
| AMTCM6I | Obtained from CSA | 0K | 0.0% | 0M | 0.0% [2] |
| AMTCM7I | Allocated | 1,122K | 61.8% | 10M | 56.0% [3] |
| AMTCM8I | High-water mark | 1,238K | 68.2% | 11M | 59.8% [4] |
| AMTCM9I | Available | 694K | 38.2% | 8M | 44.0% [5] |

**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
4. highest utilization level since the system was IPLed

5. total amount of storage available to be allocated

---

**NOTE**

The output provided to MainView AutoOPERATOR does not use commas in the numbers. For example, 2,908K appears as 2908K.

---

**CSMON**

The CSMON service will be discontinued in the next release of the product. The service is no longer required because CSMON is bundled with SYSPROG Services.

**CTCB**

The CTCB (CTC) service provides the ability to abnormally terminate any task in an 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.
## Syntax

The parameters are defined as follows:

- **tcbadrs**: specify 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.

- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information.

  This parameter is optional if you enter a symbolic name in the first parameter position that also defines the address space; otherwise, it is required.

- **U nnnn**: *(optional)* 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.

- **S xxx**: *(optional)* 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 *S xxx*, the one entered last will take precedence.

- **RSN nnnn**: *(optional)* 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.

<table>
<thead>
<tr>
<th>tcbadrs</th>
<th>- Asi</th>
<th>- Unnnn</th>
<th>- Sxxx</th>
<th>- RSN nnnn</th>
<th>DUMP</th>
<th>RETRY</th>
<th>STEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CTCB

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 @TCB004. 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,@TCB004

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,@TCB004,U2000,RSN30

DUMP (optional) indicates that a dump is requested

Note: 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.

RETRY (optional) indicates that the task’s recovery routines should be allowed to retry; retry will not be allowed if RETRY is not specified.

STEP (optional) indicates that the jobstep should also be terminated if the task fails to recover.
Example 4

You want to abnormally terminate a task. You know the TCB address, and you know that the ASID from 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 displays deviation from a list of jobs that must be active during specified time intervals and notifies you when a job that should be active is not. This list of jobs resides in a CONFIGxx member of a Logical Parmlib data set.

Syntax

The parameters are defined as follows:

|-- DEVIATN
   |   00
   |   xx

00  default suffix of the CONFIGxx member
xx  suffix of the CONFIGxx member in SYS1.PARMLIB that is to be used

The entries in the CONFIGxx member must begin in column 1. The format of the entries is

*/NAME=jobname TIME=hhmm-hhmm

The parameters are defined as follows:

jobname  job name (or Started Task ID for Started Tasks)
hhmm-hhmm beginning and ending times for the interval in which the specified job must be active

- Valid values for hh are 00 to 23.
- Valid values for mm are 00 to 59.
Example

Assume that warning messages are to be issued if job INVENTORY is not active between 8:00 A.M. and 5:00 P.M. Place the following entry in the CONFIG00 member of SYS1.PARMLIB:

*/NAME=INVENTORY TIME=0800-1700

To display deviation from the CONFIG00 member, type

deviatn 00

AMTCF61 NO DEVIATION FOUND FROM CONFIG00

If job INVENTORY is not active during the time specified in the CONFIG00 member, a warning message is displayed, as illustrated here:

deviatn 00

AMTCF81 WARNING INVENTORY SHOULD BE ACTIVE 0800-1700
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]...}
```

The parameter is defined as follows:

`nnn` facility number that is listed under the heading “Facility Indicators” in chapter 4 (Control) of IBM Publication *z/Architecture Principles of Operation*

**Example 1**

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

```
dfl 44
```

<table>
<thead>
<tr>
<th>AMTDF1I</th>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTDF2I</td>
<td>----</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>AMTDF4I</td>
<td>44</td>
<td>Perform Floating-Point Operation (PFPO) facility is installed</td>
</tr>
<tr>
<td>AMTDF5I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
</tr>
</tbody>
</table>

**Example 2**

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

```
dfl 40,67,68
```
### Example 3

To display a list of the installed features, type

```
dfl
```

<table>
<thead>
<tr>
<th>AMTDFl</th>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTDFl</td>
<td>40</td>
<td>Set-program-parameter facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>67</td>
<td>CPU-measurement counter facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>68</td>
<td>CPU-measurement sampling facility</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTDFl</th>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTDFl</td>
<td>0</td>
<td>&quot;N3&quot; instructions feature</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>1</td>
<td>z/Architecture mode installed</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>2</td>
<td>z/Architecture mode is active</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>3</td>
<td>DAT Enhancement Facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>4</td>
<td>Invalidate DAT Table Entry (IDTE) select 1</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>6</td>
<td>ASN Reuse Facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>7</td>
<td>Store Facility List Extended Feature</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>8</td>
<td>Enhanced DAF facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>9</td>
<td>Sense-Running Status Facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>10</td>
<td>Conditional SSKE Feature</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>11</td>
<td>Configuration-Topology Feature</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>16</td>
<td>Extended-Translation Facility 2</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>17</td>
<td>Message Security Assist Feature</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>18</td>
<td>Long-displacement Facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>19</td>
<td>Long-displacement Facility high performance</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>20</td>
<td>The HFP-Multitply and Subtract facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>21</td>
<td>Extended-immediate facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>22</td>
<td>Extended-translation facility 3</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>23</td>
<td>HFP-unnormalized-extension facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>24</td>
<td>ETF2-enhancement facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>25</td>
<td>Store-clock-fast facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>26</td>
<td>Parsing-enhancement facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>27</td>
<td>Move-with-optional-specifications facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>28</td>
<td>TOD-clock-steering facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>30</td>
<td>ETF3-enhancement facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>31</td>
<td>Extract-CPU time facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>32</td>
<td>Compare-and-swap-and store facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>33</td>
<td>Compare-and-swap-and store facility 2</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>34</td>
<td>General-instructions-extension facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>35</td>
<td>Execute-extensions facility</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>41</td>
<td>Floating-point-support-enhancement facilities</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>42</td>
<td>Decimal-floating-point facility (DFP)</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>43</td>
<td>DFP high performance</td>
</tr>
<tr>
<td>AMTDFl</td>
<td>44</td>
<td>Perform Floating-Point Operation (PFPO) facility</td>
</tr>
</tbody>
</table>

AMT001A SYSPROG
DONTSWAP

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

**Syntax**

```
DONTSWAP @asi
```

The parameters are defined as follows:

- `@` specifies your own address space
- `asi` address space instance; see “Identifying address spaces” on page 17 for more information

**NOTE**

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

**Example**

To make address space INVENTRY nonswappable, type

```
dontswap inventry
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS51I JOB0205 INVENTRY STEP1 Service Class=TSONRM</td>
<td></td>
</tr>
<tr>
<td>AMTS52I INVENTRY HAS BEEN SET NONSWAPPABLE (01)</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

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, including

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

**Syntax**

```
DSNAME -dsname
```

The parameter is defined as follows:

`dsname`  data set name

**Examples**

To display information about data set SYS1.DUMP05, type
dsname sys1.dump05

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

dsname master.tape

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

```
DSSUM [hdrname] COMMON
```

```
### Example 1

**dssum**

<table>
<thead>
<tr>
<th>AMTSM4I</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Max Common Area Data Spaces in use= 52.00%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Common Area Data Spaces= 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common Area Data Spaces (CADS) high-water mark= 53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common Area Data Spaces (CADS) in use= 52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of lost CADS ASTES= 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AMTSM5I**

Table sorted by COMMON

<table>
<thead>
<tr>
<th>Jobname</th>
<th>ASID</th>
<th>JITOKEN</th>
<th>All</th>
<th>Common</th>
<th>Single</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>MASTER</em></td>
<td>0001</td>
<td>:0012#BT</td>
<td>39</td>
<td>8</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>EYUX410</td>
<td>0084</td>
<td>:0402#X6</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>DC$RTCS</td>
<td>0064</td>
<td>:02M2#00</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>OMVS</td>
<td>000F</td>
<td>:00F2#BW</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>000A</td>
<td>:00A2#BT</td>
<td>62</td>
<td>3</td>
<td>11</td>
<td>76</td>
</tr>
<tr>
<td>XCFAS</td>
<td>0006</td>
<td>:0062#BW</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>JES2AUX</td>
<td>0041</td>
<td>:0102#C6</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>MTABBISS</td>
<td>00BB</td>
<td>:04V2#X$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>AAODL71</td>
<td>0172</td>
<td>:09J3U2I</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ILDTPAS5</td>
<td>0050</td>
<td>:0223F2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DEVMAN</td>
<td>000E</td>
<td>:00E2#BW</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DC$TCPID</td>
<td>009E</td>
<td>:0422#TY</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SMCO</td>
<td>0025</td>
<td>:00#2#NL</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AAONS72</td>
<td>015C</td>
<td>:08$3RO1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DC$NFSC9</td>
<td>0033</td>
<td>:01C2#N@</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ANTAS000</td>
<td>000D</td>
<td>:00D2#C6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SVOSP</td>
<td>0076</td>
<td>:0312#RU</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Example 2

dsum asid

<table>
<thead>
<tr>
<th>Jobname</th>
<th>ASID</th>
<th>JITOKEN</th>
<th>All</th>
<th>Common</th>
<th>Single</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>MASTER</em></td>
<td>0001</td>
<td>:001RIU1</td>
<td>38</td>
<td>8</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>RASP</td>
<td>0003</td>
<td>:003RIU1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>DUMPSRV</td>
<td>0005</td>
<td>:005RIU3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>XCFAS</td>
<td>0006</td>
<td>:006RIU2</td>
<td>8</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>SMSVSAM</td>
<td>0009</td>
<td>:009RIU2</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>000A</td>
<td>:00ARIU1</td>
<td>34</td>
<td>3</td>
<td>11</td>
<td>48</td>
</tr>
<tr>
<td>ANTRAM</td>
<td>000C</td>
<td>:00CRIU3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ANTAS000</td>
<td>000D</td>
<td>:00DRIUD</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DEVMAN</td>
<td>000E</td>
<td>:00ERIU3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>OMSV</td>
<td>000F</td>
<td>:00FRIU2</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>JESXCF</td>
<td>0012</td>
<td>:00IRIU3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>IOSAS</td>
<td>0015</td>
<td>:00LRIU3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>IXGLOGR</td>
<td>0016</td>
<td>:00MRIU3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
The DUMP (DUM) service displays the contents of storage in hexadecimal and character formats.

Syntax

```
DUMP  expression  length  asi
```

The parameters are defined as follows:

- **expression**: see “Expressions” on page 18
- **length**: (optional) 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 18 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.

- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information

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.

- **LIST**: displays the predefined and user-defined symbols, along with their respective definitions, that can be used with DUMP (see “EQUATE” on page 221 for information about defining symbols)
Notes

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?

Example 2

dum,CVT
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

4. sixteen-byte hexadecimal dump per line

**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
```

<table>
<thead>
<tr>
<th>AMTC1IHI USER DEFINED SYMBOLS ARE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1GI MOBJ0001 00000200_00000000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0002 00000001_00000000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0003 00000001_08200000</td>
</tr>
<tr>
<td>AMTC1GI @TCB001 7FD968</td>
</tr>
<tr>
<td>AMTC1GI @TCB001 7CAD90</td>
</tr>
<tr>
<td>AMTC1GI @TCB003 7CA930</td>
</tr>
<tr>
<td>AMTC1FI @TCB004 7BCE88</td>
</tr>
<tr>
<td>AMTC1GI @TCB005 7CA5F0</td>
</tr>
<tr>
<td>AMTC1GI @TCB006 7EE680</td>
</tr>
<tr>
<td>AMTC1GI @TCB007 7EEA40</td>
</tr>
<tr>
<td>AMTC1GI @TCB008 7EE468</td>
</tr>
<tr>
<td>AMTC1GI @TCB009 7EE150</td>
</tr>
</tbody>
</table>

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

```
dump list
```

<table>
<thead>
<tr>
<th>AMTC1IHI USER DEFINED DUMP SERVICE LABELS ARE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1I @PWVT 00008C88</td>
</tr>
<tr>
<td>AMTC1I @AVT 00009000</td>
</tr>
<tr>
<td>AMTC1I @BBCT @AVT+58?</td>
</tr>
<tr>
<td>AMTC1F PREDEFINED DUMP SERVICE CONTROL BLOCK LABELS ARE:</td>
</tr>
<tr>
<td>AMTC1I @ACB 10%+100?+14?+18?</td>
</tr>
<tr>
<td>AMTC1I @AMCBS 10%+100?</td>
</tr>
<tr>
<td>AMTC1I @ASCB 224%</td>
</tr>
<tr>
<td>AMTC1I @ASMVT 10%+2C0?</td>
</tr>
<tr>
<td>AMTC1I @JSTCB 224%+6C%+8%+7C%</td>
</tr>
</tbody>
</table>
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

\texttt{dump,jstcb+2c?,20,jes2}

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

\texttt{dump,*?,20*}

The first asterisk in the command represents the beginning of the area previously displayed, and the second asterisk indicates that the target address space is the same as in the previous command.

To display the next CDE in the chain, simply repeat the previous command by entering an asterisk; for example, type

\texttt{dump,*?,20*}

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1XI MOBJ0009+90F0</td>
<td>Size= 4095M, Guard= 1M</td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00000001_001090F8</td>
<td>KEY= 0</td>
<td>DH02DBM1 (ASID 009E)</td>
</tr>
<tr>
<td>AMTC1ZI 001090F8</td>
<td>00F20030 00000000</td>
<td><em>.........................</em></td>
</tr>
<tr>
<td>AMTC1ZI 00109100 00000000 01000000 00000001 22903000</td>
<td><em>...................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109110 00000000 00000000 00000000 143B8FE4</td>
<td><em>..........................U</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109120 BCE2BF0E AF590A68</td>
<td>C6C2D3D2 0F87E570</td>
<td><em>.S......FBLK.gV.</em></td>
</tr>
<tr>
<td>AMTC1ZI 00109130 00000001 00109000 00000000 00000000</td>
<td><em>.........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109140 00F20030 00000000 00000000 004E2000</td>
<td><em>.2..................+.</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109150 00000001 23903000 00000000 00000000</td>
<td><em>.........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109160 00000000 143B905A BCE2BF0E AF5A0BE8</td>
<td><em>..................S......Y</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109170 C6C2D3D2 00000000 00000001 00109000</td>
<td><em>FBLK.............</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109180 00000000 00000000 00000001 00109100</td>
<td><em>..........................J.</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 00109190 C6D9C5C5 00000000 00000000 00000000</td>
<td><em>FREE..............</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091A0 00000000 00000000 00000000 00000000</td>
<td><em>..........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091B0 00000000 00000000 00000000 00000000</td>
<td><em>..........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091C0 00000001 00109000 00000000 00000000</td>
<td><em>..........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091D0 00000001 00109218 C6D9C5C5 00000000</td>
<td><em>....k.FREE....</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091E0 00000000 00000000 00000000 00000000</td>
<td><em>..........................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC1ZI 001091F0 00000000 00000000 00000000 00000000</td>
<td><em>..........................</em></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

The resulting storage is within a memory object.

See “OLIST” on page 312 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 INVENTRY, where INVENTRY is an address space instance.

**NOTE**

The DVIEW service does not support the extended area of Extended Addressability Volume (EAV) devices.

The DVIEW (DVI) service displays data blocks from any online DASD volume. DVIEW assists in locating the data blocks to be displayed by searching the VTOC and PDS directories for a specified data set name and member. You also can use DVIEW to display

- extents of a data set
- DSCB of a data set
- list of data sets on the volume
- a volume’s VTOC
- a volume’s label

**Syntax**

The parameters are defined as follows:
CCHHR displays a data block using its cylinder, track, and record number

volser specifies the volume serial number of the DASD volume from which the data block is displayed

If this field is null, volser is obtained from the catalog.

address specifies the cylinder, head, and record of the data block to be displayed; the format of the operand is a 10-digit hexadecimal value, cccchhhhrr

DSN displays the data blocks requested from a specified data set or member
dsname specifies the data set name to be displayed; use the special name *FMT4 to specify the VTOC

Note: Enclose *FMT4 in single quotation marks (‘*FMT4’) when specifying this field from the SYSPROG Services selection panel.

member (optional) part of the dsname that specifies the member of a partitioned data set from which the data blocks are displayed

#skip (optional) specifies the number of data blocks to be skipped; the default value is zero (an EOF block does not stop the operation and is counted as a block)

#count (optional) specifies the number of data blocks to be displayed; the default value is one (an EOF block stops the operation)

DSCB displays the Format 1 DSCB of a specified data set

EXTENTS displays an extent list for the specified data set

FILES lists the file names in the VTOC that begin with a partial data set name

VTOCENTRY displays the VTOC entry for the specified data set

LABEL displays the label data block of a volume

NOTE
You can abbreviate the first operand to an unambiguous character. For example, you can abbreviate CCHHR to C because no other operands in the DVIEW syntax diagram begin with that letter. However, you cannot abbreviate DSCB to D or even DS because these abbreviations conflict with the DSN operand; instead, abbreviate DSCB to DSC.
Examples

The following examples illustrate ways to invoke the DVIEW service.

**CCHHR example**

To display the data block found on volume PMG05 at cylinder 40, track 13, record 26, type

dview cchhr pmg05 0028000d1a
**DSN example**

To display the first data block of data set USER.DATA on volume PMG005, type

dview dsn pmg005 user.data
### DSCB example

To display the Format 4 DSCB on volume PMG005, type

*dview dscb pmg005 *fmt4*

<table>
<thead>
<tr>
<th>AMTVD081</th>
<th>DISK VIEW SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVD02I</td>
<td>VOL - PMG005</td>
</tr>
<tr>
<td>AMTVD03I</td>
<td>BLOCK CCHHR - 0084000001</td>
</tr>
<tr>
<td>AMTVD06I</td>
<td>CSW - 0008C248</td>
</tr>
<tr>
<td>AMTVD11I</td>
<td>ECB - 4102C78C</td>
</tr>
<tr>
<td>AMTVD08I</td>
<td>SENSE - 0000</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0000 04040404 04040404 04040404 04040404 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0010 04040404 04040404 04040404 04040404 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0020 04040404 04040404 04040404 F4008600 ..................4...*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0030 1D2F0FB8 022B0000 00968901 00000230 ..................oi.....*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0040 001E4B36 010B5209 02002F24 00000000 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0050 00000000 00000000 00000000 00000000 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0060 00000000 00000000 00010000 B4000000 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0070 B6001000 00000000 00000000 00000000 ..................*</td>
</tr>
<tr>
<td>AMTVD07I</td>
<td>0080 00000000 00000000 00000000 ..................*</td>
</tr>
</tbody>
</table>

### EXTENTS example

To display an extent list of the data set USER.DATA on volume PMG005, type

*dview extents pmg005 user.data*

<table>
<thead>
<tr>
<th>AMTVD081</th>
<th>DSN - USER.DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVD25I</td>
<td>FLAGS START CCHH END CCHH</td>
</tr>
<tr>
<td>AMTVD27I</td>
<td>0100 01D00004 01D2000F</td>
</tr>
<tr>
<td>AMTVD27I</td>
<td>0101 000C000F 000C0010</td>
</tr>
</tbody>
</table>

### FILES example

To list the names of all data sets on volume PMG005 that begin with USER, type

*dview files pmg005 user*

<table>
<thead>
<tr>
<th>AMTVD14I</th>
<th>USER.TEXT.CNTL DSCB - 00840000016</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVD14I</td>
<td>USER.TEXT.CLIB DSCB - 00840000221</td>
</tr>
<tr>
<td>AMTVD14I</td>
<td>USER.TEXT.TEXT DSCB - 0084000303</td>
</tr>
</tbody>
</table>
**VTOCENTRY example**

To display the VTOC entry for the data set USER.DATA on volume PMG005, type

```
dview vtocentry pmg005 user.data
```

**LABEL example**

To display the label data block of volume PMG005, type

```
dview label pmg005
```
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:

- **classname**
  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.

- **unitname**
  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 **unitname**:

- **ALL**
  displays information for all devices

- **PUBLIC**
  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

- **PRIVATE**
  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

- **STORAGE**
  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
**Example 1**

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

```plaintext
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>3330</td>
<td>1</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>VIO3390</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>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>AMTED5I</td>
<td>DVN Dev Type VOLSER LUV Status</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>8519 3390 HSM408 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851A 3390 HSM409 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851B 3390 SPLB23 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851C 3390 SPLC25 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851D 3390 BAB339 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851E 3390 BAB340 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>851F 3390 BAB341 3010200F Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>8319 3380 BAB200 3010200E Private</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>831A 3380 BAB201 3010200E Private</td>
<td></td>
</tr>
</tbody>
</table>

10 devices were selected

AMTED8I 10 devices were selected

AMT001A SYSPROG

### 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. 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 3

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

```
editinfo, hdsbase, both
```

<table>
<thead>
<tr>
<th>AMTED4I</th>
<th>32 Devices for Generic unitname HDSBASE</th>
<th>[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTED9I</td>
<td>Mount</td>
<td></td>
</tr>
<tr>
<td>AMTED5I</td>
<td>DVN</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>Dev Type</td>
<td>Volser</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6200 3390 EM6200 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6201 3390 EM6201 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6202 3390 EM6202 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6203 3390 EM6203 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6204 3390 EM6204 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6205 3390 EM6205 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6206 3390 EM6206 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6207 3390 EM6207 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6208 3390 EM6208 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6209 3390 EM6209 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620A 3390 EM620A 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620B 3390 EM620B 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620C 3390 EM620C 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620D 3390 EM620D 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620E 3390 EM620E 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>620F 3390 EM620F 3010200F Private Online</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6300 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6301 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6302 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6303 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6304 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6305 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6306 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6307 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6308 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>6309 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630A 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630B 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630C 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630D 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630E 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED6I</td>
<td>630F 3390 3010200F offline</td>
<td></td>
</tr>
<tr>
<td>AMTED8I</td>
<td>32 devices were selected</td>
<td></td>
</tr>
<tr>
<td>AMTO01A</td>
<td>SYSPROG</td>
<td></td>
</tr>
</tbody>
</table>
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*)

| AMTED4I 18,304 Devices for Generic unitname SYSALLDA |
| AMTED7I Displaying only: online private public storage |
| AMTED01 |
| AMTED91 |
| AMTED5I DVN Dev Type VOLSER LUV Status |
| AMTED6I 8515 3390 BAB351 3010200F Private |
| AMTED6I 8521 3390 BAB350 3010200F Private |
| AMTED6I 8523 3390 BAB352 3010200F Private |
| AMTED6I 8529 3390 BAB353 3010200F Private |
| AMTED6I 852A 3390 BAB354 3010200F Private |
| AMTED6I 852B 3390 BAB355 3010200F Private |
| AMTED6I 852E 3390 BAB358 3010200F Private |
| AMTED6I 8555 3390 BAB359 3010200F Private |
| AMTED6I 8596 3390 BAB356 3010200F Private |
| AMTED6I 8597 3390 BAB357 3010200F Private |
| AMTED8I 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

edtinfo, 3390, STORAGE

<table>
<thead>
<tr>
<th>AMTED4I</th>
<th>17,320 Devices for Generic unitname 3390</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTED7I</td>
<td>Displaying only: online storage</td>
</tr>
<tr>
<td>AMTED9I</td>
<td>Mount</td>
</tr>
<tr>
<td>AMTED5I</td>
<td>DVN Dev Type VOLSER LUV Status</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>8398 3390 STRA04 3010200F Storage</td>
</tr>
<tr>
<td>AMTED6I</td>
<td>8399 3390 STRBC1 3010200F Storage</td>
</tr>
<tr>
<td>AMTED8I</td>
<td>2 devices were selected</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
</tr>
</tbody>
</table>

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

```
ENCLAVE parm1 parm2
```

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 may be modified.

Example 1

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

```
enclave
```

<table>
<thead>
<tr>
<th>AMTEN1I Owner Service</th>
<th>AMTEN2I Enclave Token Stat ASID Job Name Sys Name Class Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTEN3I</td>
<td>AMTEN4I 0000003C00000008 IL 0094 DC$TCPD2 SJSD SYSOTHER ENC#1</td>
</tr>
<tr>
<td>AMTEN4I 0000003400000006 IL 008D DC$TCPID SJSD SYSOTHER ENC#2</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000006400000016D IL 010C WS70BSS SJSD WEBMED ENC#3</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000002400000002 DL 0017 AXR SJSD STCNRM</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000002800000003 DL 002D DC$PAS SJSD STCPAS</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000003800000007 DL 0094 DC$TCPD2 SJSD SYSSTC</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000003000000005 DL 008D DC$TCPID SJSD SYSSTC</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000006800000379 DL 0044 DMRRRS01 SJSD BATLOW</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000002000000001 DL 0007 GRS SJSD SYSTEM</td>
<td></td>
</tr>
<tr>
<td>AMTEN4I 0000002C00000004 DL 0034 MIMGR SJSD SYSTEM</td>
<td></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
```

Example 3

To display the Quiesce enclave from the ENC lable, type

```
quiesce, enclave enc#1
```
The ENQUEUES (ENQ) service displays current enqueue conflicts or requested system enqueue information. 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 39.

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:

- **minorName**: R name used by ENQUEUES
- **minorName***: indicates a partial minor name; partial names can be specified by using a wildcard character (asterisk)
- **majorName**: Q name used by ENQUEUES; SYSDSN is the default
- **dsname**: data set name
- **member**: member name
- **SPFEDIT**: major name used by ISPF EDIT

---

**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.
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

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQ1PI SYSTEM (LOCAL) Q=SYSZVVDS R=CATALOG.ICFCAT.SYSC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1QI SYSID JOBNAME ASID STAT TYP TIME 14:12:19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1RI SYSB QA7A (005B) OWNS EXC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1RI SYSA FLN1 (0096) WAIT EXC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1RI SYSB QA7 (0352) WAIT EXC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1LI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1PI SYSTEM (LOCAL) Q=SYSIGGV2 R=ICFUCAT.VTSG304</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1QI SYSID JOBNAME ASID STAT TYP TIME 14:12:19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1RI SYSB ARG1 (00A2) WAIT EXC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[10] [11] [12] [13] [14]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTQ1RI SYSC MPP1 (0121) OWNS SHR RES=003 PEND 291 TSG304 NR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```
enqueues sys1.l*
```

```
AMTQ1P1 SYSTEM  (LOCAL) O=SYSDSN  R=SYS1.LPALIB
AMTQ1QI SYSID  JOBNAME   ASID   STAT TYP TIME 16:05:20
```
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
```

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:

```
enqueue * , syszvvds
```

**NOTE**

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

```
EQUATE symbol
LIST definition
DELETE length asi
```

The parameters are defined as follows:

- **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 17 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

AMTEQ7I  @JSCB=224%+6C%+8%+7C%+B4%,380

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

equate,@a,@jscb+DC?,64n

AMTEQ9I SYMBOL DEFINED

To delete symbol @A, type

equate,@a,delete

AMTEQ4I @ADELETED
ESCLASS

The ESCLASS (ESC) service provides the ability to:

- display a list of all current service classes
- display a specific service class
- change a service class
- quiesce an address space
- resume a job that has been quiesced

Syntax

```
ESC asi <newclass>
  |    QUIESCE
  |    RESUME
```

The parameters are defined as follows:

- `asi` - address space instance; see “Identifying address spaces” on page 17 for more information
- `newclass` - new service class name to be applied to the address space
- `QUIESCE (Q)` - address space is to be quiesced
- `RESUME (R)` - service class for the specified address space is to be set according to the current service policy
**Example 1**

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

esc

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPCHOT</td>
<td>APPC Hot Transactions</td>
</tr>
<tr>
<td>APPCNRM</td>
<td>APPC Normal Transactions</td>
</tr>
<tr>
<td>BATHOT</td>
<td>Batch Hot Jobs</td>
</tr>
<tr>
<td>BATNRM</td>
<td>Batch Normal Jobs</td>
</tr>
<tr>
<td>BATPROD</td>
<td>Batch Production Jobs</td>
</tr>
<tr>
<td>CICSHOT</td>
<td>CICS Hot Transactions</td>
</tr>
<tr>
<td>CICSNRM</td>
<td>CICS Normal Transactions</td>
</tr>
<tr>
<td>CICST1</td>
<td>T1** TRANS for J. Barnard</td>
</tr>
<tr>
<td>CICST2</td>
<td>T2** TRANS for J. Barnard</td>
</tr>
<tr>
<td>CICST3</td>
<td>T3** TRANS for J. Barnard</td>
</tr>
<tr>
<td>CICST4</td>
<td>T4** Trans for J. Barnard</td>
</tr>
<tr>
<td>CICST5</td>
<td>T5** TRANS for J. Barnard</td>
</tr>
<tr>
<td>COMPOSIT</td>
<td>composite service class</td>
</tr>
<tr>
<td>GRS</td>
<td>GRS Service Class</td>
</tr>
<tr>
<td>IMSHOT</td>
<td>IMS Hot Transactions</td>
</tr>
<tr>
<td>IMSNRM</td>
<td>IMS Normal Transactions</td>
</tr>
<tr>
<td>OMVSNRM</td>
<td>Open MVS Normal Transactions</td>
</tr>
<tr>
<td>RMF</td>
<td>rmf extractor service class</td>
</tr>
<tr>
<td>RMFGAT</td>
<td>RMF III Gatherer Service Class</td>
</tr>
<tr>
<td>SERVERS</td>
<td>Service class for Servers</td>
</tr>
<tr>
<td>STCLOW</td>
<td>Low Priority STC's</td>
</tr>
<tr>
<td>STCNONE</td>
<td>stc no service</td>
</tr>
<tr>
<td>STCNRM</td>
<td>Normal STC's</td>
</tr>
<tr>
<td>STCPAS</td>
<td>PAS STC's</td>
</tr>
<tr>
<td>STCPROD</td>
<td>Production STC's</td>
</tr>
<tr>
<td>STCSYS</td>
<td>System STC's</td>
</tr>
<tr>
<td>SWAPOUT</td>
<td>no service</td>
</tr>
<tr>
<td>TSOAVG</td>
<td>Normal TSO Users</td>
</tr>
<tr>
<td>TSOBBV</td>
<td>TSO BBV Users</td>
</tr>
<tr>
<td>TSOQ2</td>
<td>q/a TSO Users</td>
</tr>
<tr>
<td>TSOTEST</td>
<td>MMR Test TSO Service Class</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>HIGH PRIORITY SYSTEM WORK</td>
</tr>
</tbody>
</table>
Example 2

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

```
esc,xtshpas
```

<table>
<thead>
<tr>
<th>AMTES1I Job name:</th>
<th>XTSTHPAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTES1I Service class:</td>
<td>STCPAS</td>
</tr>
<tr>
<td>AMTES1I Description:</td>
<td>PAS STC's</td>
</tr>
<tr>
<td>AMTES1I Resource group:</td>
<td>PASSTC</td>
</tr>
<tr>
<td>AMTES1I Current period:</td>
<td>1</td>
</tr>
<tr>
<td>AMTES2I Importance lvl:</td>
<td>3</td>
</tr>
<tr>
<td>AMTES2I Period type:</td>
<td>Velocity goal</td>
</tr>
<tr>
<td>AMTES2I Goal:</td>
<td>60%</td>
</tr>
</tbody>
</table>

Example 3

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

```
esc,xtshpas,stcnrm
```

<table>
<thead>
<tr>
<th>AMTES3I Service class changed for address space XTSTHPAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTES0I</td>
</tr>
<tr>
<td>AMTES1I Job name: XTSTHPAS</td>
</tr>
<tr>
<td>AMTES1I Service class: STCNRM</td>
</tr>
<tr>
<td>AMTES1I Description: Normal STC's</td>
</tr>
<tr>
<td>AMTES1I Current period: 1</td>
</tr>
<tr>
<td>AMTES2I Importance lvl: 4</td>
</tr>
<tr>
<td>AMTES2I Period type: Velocity goal</td>
</tr>
<tr>
<td>AMTES2I Goal: 30%</td>
</tr>
</tbody>
</table>
**Example 4**

To quiesce a specified address space, type

\[ \text{esc,xtsthpas,q} \]

**Example 5**

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

\[ \text{esc,xtsthpas,resume} \]

**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
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:

- **memberName**: member within the specified or default data set
- **dsname**: specified data set name

- 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

\texttt{exe, shift1}

\begin{tabular}{l}
AMTK11I V 01A0,OFFLINE \[1] \\
AMTC22I IEE303I 01A0 OFFLINE \[2] \\
AMTK10I \\
AMTK11I V 01A1,OFFLINE \[1] \\
AMTC22I IEE202I 01A1 OFFLINE \[2] \\
AMTK10I \\
AMTK11I V 01A2,OFFLINE \[1] \\
AMTC22I IEE303I 01A2 OFFLINE \[2] \\
AMTK10I \\
AMTK11I V 01A3,OFFLINE \[1] \\
AMTC22I IEE303I 01A3 OFFLINE \[2] \\
AMTK10I
\end{tabular}

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

```plaintext
exe,status,vam3.res.lib
```

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**

```
EXIT  asi
```

The parameter is defined as follows:

\[ asi \]  
address space instance; see “Identifying address spaces” on page 17 for more information

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,$wait

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 -key
```

The parameters are defined as follows:

- `address`: address of storage to FREEMAIN, in hexadecimal format
- `length`: length of storage to FREEMAIN, in decimal format
- `spid`: storage subpool identifier: 226, 227, 228, 231, 239, 241, or 245
- `key`: key of storage: 0-15 for subpools 227, 228, 231, and 241

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

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 ddname @ asi
```

The parameters are defined as follows:

- **member**: member name
  - When aliases are used, the true member name appears in the output.
- **ddname**: DDNAME of the desired data set concatenation to be searched, or LNKLST to search the address space’s LNKLST
- **@**: your own address space
- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information
Examples

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

\texttt{find,mv,sysproc}

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTFM1I</td>
<td>JOBNAME: BMVDWP6</td>
<td>DDNAME: SYSPROC</td>
</tr>
<tr>
<td>AMTFMV1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTFM2I</td>
<td>CT</td>
<td>--Last Update--</td>
</tr>
<tr>
<td>AMTFML1</td>
<td>1</td>
<td>21Dec2004 18:21</td>
</tr>
<tr>
<td>AMTFML1</td>
<td>13</td>
<td>9Mar2004 15:08</td>
</tr>
</tbody>
</table>

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

\texttt{find,BLDQS, LNKLST,BMVDWP4}

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTFMCI</td>
<td>Searching LNKLST Set</td>
<td>ipl</td>
<td></td>
</tr>
<tr>
<td>AMTFM1I</td>
<td>Jobname: BMVDWP4</td>
<td>DDNAME: LNKLST</td>
<td>Member: BLDQS</td>
</tr>
<tr>
<td>AMTFMV1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTFM3I</td>
<td>CT</td>
<td>SIZE</td>
<td>ALIAS OF</td>
</tr>
<tr>
<td>AMTFML1</td>
<td>30</td>
<td>01908</td>
<td>ILBOQSU</td>
</tr>
<tr>
<td>AMTFML1</td>
<td>65</td>
<td>01840</td>
<td>ILBOQSU</td>
</tr>
<tr>
<td>AMTFML1</td>
<td>78</td>
<td>01898</td>
<td>ILBOQSU</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To list all RESXA proc names available through the alternate JES2, type

\texttt{find,mv,sysproc}
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

```
HELP serviceName
```

The parameter is defined as follows:

```
serviceName  name of MainView SYSPROG service
```
Example 1

To display information about the ASM service, type

```
help asm
```

<table>
<thead>
<tr>
<th>AMTH1I</th>
<th>SERVICE</th>
<th>ASM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTH1I</td>
<td>FUNCTION</td>
<td>Lists information about auxiliary storage and paging data sets.</td>
</tr>
<tr>
<td>AMTH1I</td>
<td>SYNTAX</td>
<td>AMTH1I &gt;&gt;-ASM--------------------------------------------------------------&gt;&lt;</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I +--------------------+-----------------------------------------------+</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMTH1I</td>
<td></td>
<td>AMTH1I</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
</tr>
</tbody>
</table>
Example 2

To display a list of all MainView SYSPROG Services, type

```
help
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTH11I AEW</td>
<td>Alias of WARNING</td>
</tr>
<tr>
<td>AMTH11I APF</td>
<td>Lists, adds, deletes, changes or restores entries in APF list</td>
</tr>
<tr>
<td>AMTH11I ASM</td>
<td>Lists information about auxiliary storage &amp; paging data sets</td>
</tr>
<tr>
<td>AMTH11I ASVT</td>
<td>Displays address space vector table information</td>
</tr>
<tr>
<td>AMTH11I AUTHTSO</td>
<td>Lists, builds, and restores TSO authorization lists</td>
</tr>
<tr>
<td>AMTH11I BBXS</td>
<td>Manages BMC Software subsystem services</td>
</tr>
<tr>
<td>AMTH11I CDE</td>
<td>Lists the modules loaded in a specified address space</td>
</tr>
<tr>
<td>AMTH11I CLEAR</td>
<td>Clears system dump data sets without printing the information</td>
</tr>
<tr>
<td>AMTH11I CNSL</td>
<td>Simulates an MVS operator's console at a TSO user's terminal</td>
</tr>
<tr>
<td>AMTH11I COMMAND</td>
<td>Executes operator commands from a TSO terminal</td>
</tr>
<tr>
<td>AMTH11I CONSOLES</td>
<td>Displays MCS console buffer usage</td>
</tr>
<tr>
<td>AMTH11I CPU</td>
<td>Displays the proportionate use of CPU resources by job</td>
</tr>
<tr>
<td>AMTH11I CSA</td>
<td>Displays the level of CSA and ECSA usage</td>
</tr>
<tr>
<td>AMTH11I CSMON</td>
<td>Displays the use of common storage by address space</td>
</tr>
<tr>
<td>AMTH11I DEVIATN</td>
<td>Displays deviation from a list of jobs in member CONFIGxx</td>
</tr>
<tr>
<td>AMTH11I DFLIST</td>
<td>Displays information about optional CPU features</td>
</tr>
<tr>
<td>AMTH11I DONTSWAP</td>
<td>Makes an address space nonswappable</td>
</tr>
<tr>
<td>AMTH11I DSNAME</td>
<td>Displays the volume serial number of a cataloged data set</td>
</tr>
<tr>
<td>AMTH11I DSSUM</td>
<td>Displays Data Space information</td>
</tr>
<tr>
<td>AMTH11I DUMP</td>
<td>Displays storage in hexadecimal and character formats</td>
</tr>
<tr>
<td>AMTH11I DVIEW</td>
<td>Displays data blocks from any online DASD volume</td>
</tr>
<tr>
<td>AMTH11I EDTINFO</td>
<td>Displays Eligible Device Table (EDT) information</td>
</tr>
<tr>
<td>AMTH11I ENQUEUE</td>
<td>Displays system enqueue and enqueue conflict information</td>
</tr>
</tbody>
</table>
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

```
INFO
```

Example 1

To display information about the current operating environment, type

```
info
```

```
AMTR01I System Name=SJSD
AMTR01I SYSPROG Release 3.5.00
AMTR02I z/OS 01.06.00   FMID HBB7709  z/Architecture   [1]
AMTR03I CPU 0 Serial Number 04381A Model 2084, Type=CP   [2]
AMTR03I CPU 1 Serial Number 04381A Model 2084, Type=ZAAP
AMTR03I CPU 2 Serial Number 04381A Model 2084, Type=ZAAP
AMTR03I CPU 16 Serial Number 04381A Model 2084, Type=CP
AMTR04I LPAR name=SJSD, Max of config allowed= 11.1%
AMTR04I Logical CPUs: Active= 2, Standby= 13, Reserved= 0
AMTR04I Last IPL was COLD Start (CLPA) on 3/03/2010 at 0:25:02   [3]
AMTR04I From SAZ16I
AMTR04I IPL used LOADD0 in SYS1.IPLPARM on volume HCD002 (8505)
AMTR04I IEASYM list = D1
AMTR04I IEASYS list = (D1), Source=Oper
AMTR05I TSO Version 3 Release 6 Mod 0   [4]
AMTR06I VTAM Terminal ID  TCPBL492   [5]
AMTR06I Common Storage Tracking: CSA=ON  SQA=ON
AMTR06I
AMTR06I Volser PARMLIB Data Set Name
AMTR06I ------ ---------------------
AMTR06I        SYS1.PARMLIB
AMTR06I
AMTR06I Current LNKLST set is IPL
```
Legend:

1. operating system release and its FMID
2. serial number for each CPU and the CPU model
3. type of IPL most recently performed (cold start, warm start, quick start)
4. current release of TSO/E
5. VTAM logical terminal ID

**Example 2**

Message AMTR0PI is displayed by the INFO Service when (and only when) a Program Event Recording (PER) SLIP trap is active. When displayed, AMTR0PI is the first message produced by the INFO service. When active, PER slows down the processor.

To display information when a PER SLIP trap is active, type

```
info
```

```
AMTR0PI Program Event Recording (PER) SLIP trap DP#0 is active
AMTR0HI System Name=SJSD
AMTR02I z/OS 01.10.00 FMID HBB7750 z/Architecture HiperDispatch=ON
AMTR03I CPU 0 Serial Number 0454A2 Model 2097, Type=CP
AMTR03I CPU 1 Serial Number 0454A2 Model 2097, Type=zAAP
AMTR03I CPU 2 Serial Number 0454A2 Model 2097, Type=ziIP
AMTR0FI LPAR name=SJSD, Max of config allowed= 10.0%
AMTR0GI Logical CPUs: Active= 1, Standby= 1, Reserved= 0
AMTR04I Last IPL was COLD Start (CLPA) on 3/01/2010 at 22:20:07
AMTR0II From SAZ1AW on AC12
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 10 Mod 0
AMTR06I VTAM Terminal ID TCPD0024
AMTR0EI Common Storage Tracking: CSA=ON  SQA=ON
AMTR00I
AMTR07I Volser PARMLIB Data Set Name
AMTR08I ------ ---------------------
AMTR09I SAZ1AW SYS1.PARMLIB
AMTR00I
AMTR0AI Current LNKLST set is IPL
AMT001A SYSPROG
```
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:

- `dvn`: device number for which outstanding non-TP I/O is to be displayed
- `volser`: volume serial number for which outstanding non-TP I/O is to be displayed

---

### Example 1

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

```
io
```

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTI11I</td>
<td>INVENTORY UNIT 223 PACK08 I0Q 00F832A0 DRIVER-EXCP CCHH- 191 3</td>
<td></td>
</tr>
<tr>
<td>AMTI11I * TEST UNIT 180 333214 I0Q 00F85A00 DRIVER-VSAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[7]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTI11I + <em>MASTER</em> UNIT 1A4 PAGE11 I0Q 00F91300 DRIVER-ASM CCHH- 271 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Legend:

1. one-character flag; valid characters are as follows:
   
   blank  line represents the I/O request currently active on the device
   *   I/O request on this line has been queued by the I/O supervisor
       pending completion of a previous I/O request
   +   ASM channel program is active on this device

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>
<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>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>IO 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>IO 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>
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 I/O for volume PACK08, type

```plaintext
io pack08
```

<table>
<thead>
<tr>
<th>AMTI11I</th>
<th>GOJOB UNIT 283 PACK08 IOQ 00F83DA0 DRIVER-EXCP CCHH- 191 3 [1] [2] [3] [4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTI12I</td>
<td>UNIT B00: RESERVES-0 ALLOCATIONS-3 OPENS-2</td>
</tr>
</tbody>
</table>

**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 AMTI12I 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**

```
IPLDATA [prefix]
```

The parameter is defined as follows:

- `prefix` *(optional)* if specified, only the parameters that begin with the specified `prefix` will be displayed (see “Example 2” on page 248)

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

```
  ipl
```

```
AMTIPI I PAR=SYSC, IPL on Monday, May 10, YYYY 3:53:56:52 PM Local
Loadparm Unit=7D00, Volser=TSG316, SYS1.IPLPARM(LOADC6)
IODF 04 SYSP IODF000C1 C1
SYSCAT TSG31813CICFMCAT.SYSC(OS390260
IEASYM C6
SYSPLEX BBPLEX01 X
NUCLEUS 1
NUCLST 00 N
Parmlib Unit= 2B2, Volser=TSG321, SYS1.PARMLIB

IEASYS list=(C1)
Source Parm               IEASYSxx Parameter Value
------------------- ---------------------------------------------------
Default ALLOC           
Default APF             
IEASYSC6 APG 07         
Default BLDL            
Default BLDLF           
IEASYSC6 CLOCK 00       
Default CLPA            
IEASYSC6 CMB (UNITR,COMM,GRAPH,CHRDR)
IEASYSC6 CMD C3         
IEASYSC6 CON            
Default CONT            
IEASYSC6 COUPLE SC      
Default CPQE            
IEASYSC6 CSA (3548,300M)
IEASYSC6 CSCBLOC ABOVE  
Default CVIO             
Default DEVSUP          
IEASYSC6 DIAG SC        
IEASYSC6 DUMP DASD     
Default DUPLEX          
Default EXIT            
IEASYSC6 FIX C6         
IEASYSC6 GRS STAR       
Default GRSCNF 00       
IEASYSC6 GRSRNL 00      
IEASYSC6 ICS PT         
IEASYSC6 IOS 00         
Default IPS 00           
Default LNK 00           
IEASYSC6 LNKAUTH APFTAB 
IEASYSC6 LOGCLS L       
```
IEASYSC6 LOGLMT 999999
Default LOGREC SYS1.LOGREC
IEASYSC6 LPA (26,C6)
IEASYSC6 MAXCAD 60
IEASYSC6 MAXUSER 370
IEASYSC6 MLPA C4
IEASYSC6 MSTRJCL 00
Default NONVIO
IEASYSC6 NSYSLX 100
Default NUCMAP
IEASYSC6 OMVS C6
IEASYSC6 OPI YES
IEASYSC6 OPT PT
Default PAGE-OP
IEASYSC6 PAGE-SYS (PAGE.VPAGC31.PLPA,PAGE.VPAGC31.COMMON,PAGE.VPAGC31.LOCAL1)
Default PAGNUM
IEASYSC6 PAGTOTL (20,20)
IEASYSC6 PAK 00
IEASYSC6 PLEXCFG MULTISYSTEM
IEASYSC6 PRODP C5
IEASYSC6 PROG (26,C6,OL)
Default PURGE
Default RDE NO
IEASYSC6 REAL 128
Default RER NO
IEASYSC6 RSU 0
IEASYSC6 RSVNONR 25
IEASYSC6 RSVSTRT 25
IEASYSC6 SCH 00
IEASYSC6 SMF 00
IEASYSC6 SMS SC
IEASYSC6 SQA (6,40)
IEASYSC6 SSN SC
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`

```
Example 1

AMTIP11 LPAR=SJSD, IPL on Tuesday, April 29, YYYY at 1:03:57 AM Local
AMTIP21 Loadparm Unit=a620, Volser=HCD002, SYS1.IPLPARM(LOADDD0)
AMTIP01
AMTIP31 IODF 47 BMCS CONFIGB1 00 TREQ0
AMTIP31 SYSCAT SACATD11CCATALOG.ICF.MSTASJSD
AMTIP31 IEASYM D1
AMTIP31 SYSPLEX BBPLEX01 X
AMTIP31 NUCLEUS 1
AMTIP31 NUCLST 00 N
AMTIP01
AMTIP41 Parmlib Unit=85FD, Volser=SAZ191, SYS1.PARMLIB
AMTIP01
AMTIP51 IEASY list = (D1)
AMTIP01
AMTIP71 Source Parm IEASYxx Parameter Value
AMTIP81 -------- -------- ----------------------------------------
AMTIP91 IEASYSD1 GRS TRYJOIN
AMTIP91 Default GRSCNF 00
AMTIP91 IEASYSD1 GRSRNL EXCLUDE
AMTIP91 End of IPLDATA Service
```
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**

To display SMF data, type

```plaintext
ismf
```

---

**Example 1**

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

<table>
<thead>
<tr>
<th>AMTSICI Logstream Name=</th>
<th>IFA5MF.SJSD.LOGR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTSIEI Status=</td>
<td>Connected, Active</td>
</tr>
<tr>
<td>AMTSIFI Buffer Size=</td>
<td>65,532 bytes</td>
</tr>
<tr>
<td>AMTSIGI Last Write=</td>
<td>9:29:01.559 6/22/YYYY</td>
</tr>
<tr>
<td>AMTSI01I AMTSICI Logstream Name=</td>
<td>IFA5MF.SJSD.LOGR1</td>
</tr>
<tr>
<td>AMTSIDI SMF Record Types=</td>
<td>70:79, 240</td>
</tr>
<tr>
<td>AMTSIEI Status=</td>
<td>Connected, Active</td>
</tr>
<tr>
<td>AMTSIFI Buffer Size=</td>
<td>65,532 bytes</td>
</tr>
<tr>
<td>AMTSIGI Last Write=</td>
<td>9:15:17.030 6/22/YYYY</td>
</tr>
<tr>
<td>AMT001A SYSPROG</td>
<td></td>
</tr>
</tbody>
</table>

This example shows information displayed when SMF data is being recorded in the logstream.
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.

- **n**: 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.

- **keyword**: 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.
Example 1

Sample processor usage for one second and sort the resulting data by the column containing CP utilization in descending sequence

\texttt{jcpu,1,cp}

<table>
<thead>
<tr>
<th>Jobname</th>
<th>ASID</th>
<th>% CP</th>
<th>% zIIP</th>
<th>% zAAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTJC3I BMRSBLD1</td>
<td>008A</td>
<td>50.95</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I BMVkEM11</td>
<td>0076</td>
<td>46.76</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I XCFAS</td>
<td>0006</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I WLM</td>
<td>000B</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I BMVDP3</td>
<td>00C2</td>
<td>0.15</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$TCPIP</td>
<td>0069</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I MIMGR</td>
<td>0044</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$PAS</td>
<td>006E</td>
<td>0.12</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$CAS</td>
<td>006C</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I MSTROLM</td>
<td>009E</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I GRS</td>
<td>0007</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I JES2MON</td>
<td>001C</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$TCPAS</td>
<td>00A7</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I <em>MSTR</em></td>
<td>0001</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I SMSVSAM</td>
<td>0009</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I IMWEBRVR</td>
<td>00A4</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I MVQACAS</td>
<td>00AD</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I TN3270</td>
<td>00A5</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I TRITUNE</td>
<td>0046</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I MSTRCON</td>
<td>0028</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I CNMNETB</td>
<td>0065</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I JES2S001</td>
<td>0042</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I JES2</td>
<td>001E</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$MTCP</td>
<td>00A8</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I ULRADOPT</td>
<td>006B</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I CSTGATE</td>
<td>0030</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I RMF</td>
<td>0033</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I BSDDP1</td>
<td>00EE</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DC$BBIB</td>
<td>0021</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I ZFS</td>
<td>0026</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I CSTSACS</td>
<td>00EA</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I RACF</td>
<td>0014</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I CSTSACD</td>
<td>0043</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I INTUNE</td>
<td>004C</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC3I DEE7MSTR</td>
<td>0094</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMTJC4I</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>AMTJC5I Totals</td>
<td>-----</td>
<td>100.01</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AMT001A SYSPROG</td>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
</tr>
</tbody>
</table>
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 (:).

**NOTE**
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

```plaintext
JITOKEN  asi
```

The parameter is defined as follows:

`asi` address space instance; see “Identifying address spaces” on page 17 for more information

### Example

To display the Job Instance Token and other information for XTST6PAS, type `jit,xtst6pas`

```
AMTJT11 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]
AMTJT8I 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

4. amount of time since the job instance was initiated

5. current service class for the job instance

**JSTORM**

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**

```
JSTORM [JOBNAME] [keyword]
```

The parameters are defined as follows:
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:**

<table>
<thead>
<tr>
<th>AMTJS1</th>
<th>Jobname</th>
<th>ASID</th>
<th>Private Hex</th>
<th>SHR</th>
<th>Shared Private Hex</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTJS2</td>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-----</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>AMTJS3</td>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-----</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DC$TCPID 0092</td>
<td>0 0</td>
<td>256Gi 256Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DC$TCPD2 0099</td>
<td>0 0</td>
<td>256Gi 256Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DEENDIST 007C</td>
<td>0 0</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DIBIMSTR 0081</td>
<td>0 0</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DEENMSTR 0083</td>
<td>0 0</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DIB1DIST 0096</td>
<td>0 0</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DEENDBM1 0090</td>
<td>8,980Mi 9,062Mi</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS4</td>
<td>DIB1DBM1 0091</td>
<td>8,980Mi 8,990Mi</td>
<td>128Gi 128Gi</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTJS6</td>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-----</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>AMTJS6</td>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-----</td>
<td>--------------------</td>
<td>--------</td>
</tr>
</tbody>
</table>

---
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:

- `dvn` hexadecimal device number of the tape drive
- `EXT` causes a full 80 bytes of label information to be displayed

**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`

```
AMTL12I VOL1000111 OWNRID [1]
[2]
AMTL12I HDR1USER.DATASET 00000100010001 091775 [3]
AMTL12I HDR2F035200008031MYJOB /STEPONE [4]
```

Legend:

1. volume label
2. data set name
3. header label #1
4. header label #2
The LCPU (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**

The parameters are defined as follows:

- **asi** address space instance; see “Identifying address spaces” on page 17 for more information
- **minutes** number of minutes for the new CPU time limit
- **seconds** number of seconds for the new CPU time limit
- **U** 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

The maximum value that can be specified is 1,440.
### Examples

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

```plaintext
lcpu, vam3a
```

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLC1I</td>
<td>CPU limit for job VAM3A</td>
<td>[4] 10 minutes</td>
</tr>
<tr>
<td>AMTLC2I</td>
<td>CPU limit for current step ASMSTEP</td>
<td>[4] 2 minutes, 30 seconds</td>
</tr>
<tr>
<td>AMTLC3I</td>
<td>Step time used</td>
<td>[4] 1 minutes, 35 seconds</td>
</tr>
<tr>
<td>AMTLC4I</td>
<td>Step time remaining</td>
<td>[4] 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

```plaintext
lcpu, vam3a, 5
```

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLC1I</td>
<td>CPU limit for job VAM3A</td>
<td>[4] 10 minutes</td>
</tr>
<tr>
<td>AMTLC2I</td>
<td>CPU limit for current step ASMSTEP</td>
<td>[4] 5 minutes</td>
</tr>
<tr>
<td>AMTLC3I</td>
<td>Step time used</td>
<td>[4] 1 minutes, 35 seconds</td>
</tr>
<tr>
<td>AMTLC4I</td>
<td>Step time remaining</td>
<td>[4] 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
- 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**

```
LNLST
  STATUS
  LIST setname
  USERS setname
  PROCESS suffix1, suffix2...suffixn
  ACTIVATE setname
  UPDATE asi
  DEACT setname
  UNDEFINE setname
  LOCATE modname
```

**Note:** All function keywords (first parameter) can be abbreviated to their first three letters.

The parameters are defined as follows:

- **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.
- **PROCESS** allows you to process a LNKLST set; you must specify at least one suffix
ACTIVATE allows you to make the specified LNKLST the current set; you must specify a LNKLST set name.

UPDATE changes one or more address spaces to the current LNKLST set.

Alternatively, you can specify ALL to indicate all address spaces or a jobname-pattern containing wildcard characters.*

DEACT changes all address spaces assigned to the specified LNKLST set to the current LNKLST set.

UNDEFINE removes the specified LNKLST set from the system.

LOCATE searches the JOBLIB/STEPLIB, LPA, and the current LNKLST set for the specified load module and displays its location.

CURRENT displays currently active LNKLST set.

suffix1, suffix2... indicates that one or more member suffixes must be specified after the PROCESS keyword.

suffixn (The suffixes are appended to PROG to form the member names to be processed.)

asi address space instance; see “Identifying address spaces” on page 17 for more information.

modname limits the display to the specified module.

* 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 PROGxx 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 LNKLST set, the name of the data set in which it was found is displayed.
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**

```
<table>
<thead>
<tr>
<th>LOADLPA</th>
<th>LIST</th>
<th>USERLIB</th>
<th>DDname</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADD</td>
<td>modname</td>
<td>dsname</td>
</tr>
<tr>
<td></td>
<td>DEL</td>
<td>modname</td>
<td>ALL</td>
</tr>
</tbody>
</table>
```

The parameters are defined as follows:

**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

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.)
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`

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTL5HI</td>
<td>NAME=USERMOD, EP=00B41000, CDE=F47AD0, TIME=17:13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL5JI</td>
<td>LA=00BA4100, LEN=0022CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL5KI</td>
<td>ALIAS=USEREP1, EP=00BA4140, CDE=F47B00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL5KI</td>
<td>ALIAS=USEREP2, EP=00BA4440, CDE=F47B30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL5LI</td>
<td>LOADLPA ADD FUNCTION COMPLETED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```plaintext
loadlpa del usermod
```

---

**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 407 and “ZAP” on page 444 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:

```plaintext
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.
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

```
LOGGING xx dsname
  STOP
  FREE
  STATUS
```

The parameters are defined as follows:

- **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.

**CLASS c** 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.

**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.

**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.

**START $hhmm$** 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.

**STOP $hhmm$** 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)

If you provide both CNT and STOP statements, logging will terminate when the first condition is reached.

**HOLD** indicates that the SYSOUT output is to be held.

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.
NOHOLD indicates that the SYSOUT output is *not* to be held.

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.

MAXOUT $n$ indicates the maximum number ($n$) of lines that will be written to a SYSOUT data set.

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.

**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**

```plaintext
LWAIT -asi [hours] [minutes] [U]
```

The parameters are defined as follows:

- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information
- **hours**: number of hours for the new wait limit
- **minutes**: number of minutes for the new wait limit

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.

- **U**: in the third or fourth parameter position, indicates that the wait limit can be set to a value less than the current limit value
## 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>AMTLW1I VAM3W1</td>
<td>AMTLW4I Has been waiting for</td>
<td>4 minutes, 37 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I Wait limit remaining</td>
<td>1 hours, 55 minutes, 22 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I Wait limit is</td>
<td>2 hours, 00 minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLWoI Wait limit changed</td>
<td></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>AMTLW1I VAM3W1</td>
<td>AMTLW4I Has been waiting for</td>
<td>4 minutes, 58 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I Wait limit remaining</td>
<td>2 hours, 55 minutes, 22 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I Wait limit is</td>
<td>3 hours, 00 minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLWoI Wait limit changed</td>
<td></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>AMTLW1I Request denied. New value is less than old value.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW1I VAM3W1</td>
<td>AMTLW5I Has been waiting for</td>
<td>6 minutes, 45 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I Wait limit remaining</td>
<td>53 minutes, 14 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I Wait limit is</td>
<td>1 hours, 00 minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLWoI Wait limit changed</td>
<td></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.

\texttt{lwait,vam3w1,30,u}

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLW1I VAM3W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW4I Has been waiting for [6]</td>
<td>7 minutes, 10 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW5I Wait limit remaining [6]</td>
<td>22 minutes, 49 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW6I Wait limit is [6]</td>
<td>30 minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTLW0I Wait limit changed [7]</td>
<td></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), 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:

- `asi`: address space instance; see “Identifying address spaces” on page 17 for more information
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.

ALL keyword indicating that a display of all LXs is desired

See example 2.

**Example 1**

To display LX information for job name VLF, type

lx, vlf

<table>
<thead>
<tr>
<th>AMTLXCI</th>
<th>AMTLXDI</th>
<th>AMTLXO1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Used</td>
<td>Max</td>
</tr>
<tr>
<td>AMTLXEI</td>
<td>AMTLXFI</td>
<td>System Standard</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Standard</td>
<td>34</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System Big</td>
<td>133</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Big</td>
<td>114</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTLX7I</th>
<th>AMTLX8I</th>
<th>AMTLX9I</th>
<th>AMT001A</th>
</tr>
</thead>
<tbody>
<tr>
<td>LX Num Flags</td>
<td>ASID Jobname</td>
<td>LX Num Flags</td>
<td>ASID Jobname</td>
</tr>
<tr>
<td>AMTLX9I</td>
<td>00000A</td>
<td>001E VLF</td>
<td></td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Example 2

To display summary information, type

lx

<table>
<thead>
<tr>
<th>AMTLXCI</th>
<th>ASN and LX Reuse architecture installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXDI</td>
<td>ASN and LX Reuse architecture is enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTLX0I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>AMTLX2I Description</th>
<th>AMTLXFI System Standard</th>
<th>Used</th>
<th>Max</th>
<th>Pct Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXFI Non-Sys Standard</td>
<td>180</td>
<td>223</td>
<td>80.71%</td>
<td></td>
</tr>
<tr>
<td>AMTLXFI System Big</td>
<td>133</td>
<td>1,024</td>
<td>12.98%</td>
<td></td>
</tr>
<tr>
<td>AMTLXFI Non-Sys Big</td>
<td>114</td>
<td>29,697</td>
<td>0.38%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTLX3I</th>
<th>Dormant System LX's -</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLX4I</td>
<td>LX Num</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>AMTLX5I</td>
<td>------</td>
</tr>
<tr>
<td>AMTLX6I</td>
<td>000018</td>
</tr>
<tr>
<td>AMTLX6I</td>
<td>00003F</td>
</tr>
<tr>
<td>AMTLX6I</td>
<td>000041</td>
</tr>
<tr>
<td>AMTLX6I</td>
<td>000044</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTLX0I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>AMTLXKI</th>
<th>Top Ten LX Owners ---</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXGI</td>
<td>Job Name</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>AMTLXHI</td>
<td>---------</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td><em>MASTER</em></td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DEEOMSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DHH1MSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DIB1MSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DIO2MSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>EYUX410</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>CSQ7MSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>CSQ8MSTR</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DEE0DBM1</td>
</tr>
<tr>
<td>AMTLXJ1</td>
<td>DHH1DBM1</td>
</tr>
</tbody>
</table>
**Example 3**

To display information for LX number 10 decimal, type

lx,10

<table>
<thead>
<tr>
<th>AMTLXCI</th>
<th>ASN and LX Reuse architecture installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXDI</td>
<td>ASN and LX Reuse architecture is enabled</td>
</tr>
<tr>
<td>AMTLX0I</td>
<td></td>
</tr>
<tr>
<td>AMTLX2I</td>
<td>Description</td>
</tr>
<tr>
<td>AMTLXEI</td>
<td>Used</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System Standard</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Standard</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System Big</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Big</td>
</tr>
<tr>
<td>AMTLX0I</td>
<td></td>
</tr>
<tr>
<td>AMTLX7I</td>
<td>LX Num Flags</td>
</tr>
<tr>
<td>AMTLX8I</td>
<td>----- ----- ----- -----</td>
</tr>
<tr>
<td>AMTLX9I</td>
<td>00000A</td>
</tr>
</tbody>
</table>

**Example 4**

To display information for LX number 10 hexadecimal, type

lx,15 X'10'

<table>
<thead>
<tr>
<th>AMTLXCI</th>
<th>ASN and LX Reuse architecture installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTLXDI</td>
<td>ASN and LX Reuse architecture is enabled</td>
</tr>
<tr>
<td>AMTLX0I</td>
<td></td>
</tr>
<tr>
<td>AMTLX2I</td>
<td>Description</td>
</tr>
<tr>
<td>AMTLXEI</td>
<td>Used</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System Standard</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Standard</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>System Big</td>
</tr>
<tr>
<td>AMTLXFI</td>
<td>Non-Sys Big</td>
</tr>
<tr>
<td>AMTLX0I</td>
<td></td>
</tr>
<tr>
<td>AMTLX7I</td>
<td>LX Num Flags</td>
</tr>
<tr>
<td>AMTLX8I</td>
<td>----- ----- ----- -----</td>
</tr>
<tr>
<td>AMTLX9I</td>
<td>000010</td>
</tr>
</tbody>
</table>
MCOMMAND

(MCO) Alias for COMMAND. See the service “COMMAND” on page 178.
MDEVICE

The MDEVICE (MDE) service monitors all or selected devices attached to all processors in the configuration sampled within a 15-second interval. The MDEVICE service

- 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:

\[ dvn \]  
device number

\[ dvn1-dvn2 \]  
range of device numbers; \( dvn1 \) is the lowest device number and \( dvn2 \) is the highest device number

\[ time \]  
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).

The value is calculated as follows:

\[ \text{ServiceTime} = \text{PendingTime} + \text{DoscpmmectTime} + \text{ConnectTime} \]

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
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

MEMSCAN

- **string**
- **‘string’**
- **?**
- **?CURRENT**
- **@**
- ***CURRENT**
The parameters are defined as follows:

- **string**
  - 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.

- **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 287.

- **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” 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>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOVE</td>
<td>Causes any previously created symbols of the format <code>@nn</code> to be deleted prior to starting the scan (where <code>nn</code> 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>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>Scans only the private area of the address space within which the SYSPROG Services service is executing.</td>
<td></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, 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

\texttt{lpaliggpost0}

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,?
```

Example 3

To scan all storage for a hexadecimal string and display each occurrence, type

```
mem,x'00af72f8',dump,all
```

---

<table>
<thead>
<tr>
<th>AREA TITLE</th>
<th>START</th>
<th>END</th>
<th>MEMSCAN ABBR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSA</td>
<td>00000000</td>
<td>00001000</td>
<td>PSA</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>00001000</td>
<td>007FFFFF</td>
<td>PVT</td>
</tr>
<tr>
<td>CSA</td>
<td>00800000</td>
<td>00B97FFF</td>
<td>CSA</td>
</tr>
<tr>
<td>MLPA</td>
<td>00B98000</td>
<td>00C2FFFF</td>
<td>MLPA</td>
</tr>
<tr>
<td>PLPA</td>
<td>00C30000</td>
<td>00EBDFFF</td>
<td>PLPA</td>
</tr>
<tr>
<td>SQA</td>
<td>00EBE000</td>
<td>00FC9FFF</td>
<td>SQA</td>
</tr>
<tr>
<td>NUC R/W</td>
<td>00FCA000</td>
<td>00FD9FFF</td>
<td>NWNUC</td>
</tr>
<tr>
<td>NUC R/O</td>
<td>00FDA000</td>
<td>00FFFFFF</td>
<td>RONUC</td>
</tr>
<tr>
<td>EXT-NUC R/O</td>
<td>01000000</td>
<td>01466FFF</td>
<td>ERONUC</td>
</tr>
<tr>
<td>EXT-NUC R/W</td>
<td>01467000</td>
<td>016D1FFF</td>
<td>ERWNUC</td>
</tr>
<tr>
<td>EXTSAQ</td>
<td>016D2000</td>
<td>020AFFFF</td>
<td>ESQA</td>
</tr>
<tr>
<td>EXT-PLPA</td>
<td>020B0000</td>
<td>043B9FFF</td>
<td>EPLPA</td>
</tr>
<tr>
<td>EXT-FLPA</td>
<td>043BA000</td>
<td>043BCFFFF</td>
<td>EFLPA</td>
</tr>
<tr>
<td>EXT-MLPA</td>
<td>043BD000</td>
<td>043D3FFF</td>
<td>EMLPA</td>
</tr>
<tr>
<td>EXT-CSA</td>
<td>043D4000</td>
<td>07FFFFFF</td>
<td>ECSA</td>
</tr>
<tr>
<td>EXT-PRIVATE</td>
<td>08000000</td>
<td>7FFFFFFF</td>
<td>EPVT</td>
</tr>
</tbody>
</table>

---

```
AMTVS8I SEARCHING FOR X'00AF72F8'
AMTVSWS WILL BE SCANNING: ALL STORAGE
AMTVSNN SCANNING: PRIVATE 00001000-007FFFF
AMTVSNN SCANNING: CSA 00800000-00B97FFF
AMTVS9I STRING FOUND AT 00AF7384 IN CSA
```

```
AMTVS5I 00AF7380 +04 F1000080 00AF72F8 00AF73B8 00AFAF80 *1 8 *
AMTVS5I 00AF7390 00000000 00000000 00000000 00000000 *     *
AMTVS5I 00AF73A0 00000000 00000000 00000000 00000000 *     *
AMTVS5I 00AF73B0 00000000 00000000 00AF9140 0751FE40 *   j *
```

```
AMTVSNN SCANNING: MLPA 00B98000-00C2FFFF
AMTVSNN SCANNING: PLPA 00C30000-00EBDFFF
AMTVSNN SCANNING: SQA 00EBE000-00FC9FFF
AMTVSNN SCANNING: NUC R/W 00FCA000-00FD9FFF
AMTVSNN SCANNING: NUC R/O 00FDA000-00FFFFFF
AMTVSNN SCANNING: EXT-NUC R/O 01000000-01466FFF
AMTVSNN SCANNING: EXT-NUC R/W 01467000-016D1FFF
```
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 REFINDE` 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:

- **chpid**
  - channel path identifier
- **chpid1-chpid2**
  - range of channel path identifiers; *chpid1* is the lowest channel path identifier, and *chpid2* is the highest channel path identifier
- **time**
  - specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 1,200 seconds

**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, type

mio

<table>
<thead>
<tr>
<th>AMT030I</th>
<th>I/O data being gathered for 15 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMT031I</td>
<td>Channel Path Utilization Report</td>
</tr>
<tr>
<td>[1]</td>
<td>[2]</td>
</tr>
<tr>
<td>AMT032I</td>
<td>CHANNEL PATH 00 (BYTE) WAS 5% BUSY</td>
</tr>
<tr>
<td>AMT032I</td>
<td>CHANNEL PATH 03 (BLOCK) WAS 23% BUSY</td>
</tr>
<tr>
<td>AMT032I</td>
<td>CHANNEL PATH 12 (BLOCK) WAS 45% BUSY</td>
</tr>
<tr>
<td>AMT032I</td>
<td>CHANNEL PATH 23 (BLOCK) WAS 17% BUS</td>
</tr>
<tr>
<td>AMT033I</td>
<td>Processor I/O Interrupt Activity Report</td>
</tr>
<tr>
<td>AMT034I</td>
<td>CPU 00 Processed 4,670 Interrupts ( 6% were through TPI)</td>
</tr>
<tr>
<td>AMT035I</td>
<td>Device and Logical Control Unit Report</td>
</tr>
<tr>
<td>AMT036D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT036H</td>
<td>Dev Volser %Bsy Q1th Seek ACYL Rate Conn Pend Disct DAO</td>
</tr>
<tr>
<td>AMT036I</td>
<td>3500</td>
</tr>
<tr>
<td>AMT038D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT038H</td>
<td>ASIDname ASID %Bsy</td>
</tr>
<tr>
<td>AMT038I</td>
<td>CONSOLE 000A 100</td>
</tr>
<tr>
<td>AMT036D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT036H</td>
<td>Dev Volser %Bsy Q1th Seek ACYL Rate Conn Pend Disct DAO</td>
</tr>
<tr>
<td>AMT036I</td>
<td>B58E USSA01 0 0 0 0 .20 0 0 0 0</td>
</tr>
<tr>
<td>AMT038D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT038H</td>
<td>ASIDname ASID %Bsy</td>
</tr>
<tr>
<td>AMT038I</td>
<td>MV60DS36 014F 100</td>
</tr>
<tr>
<td>AMT036D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT036H</td>
<td>Dev Volser %Bsy Q1th Seek ACYL Rate Conn Pend DisctDAO</td>
</tr>
<tr>
<td>AMT036I</td>
<td>8370 BAB327 7 0 2 1 25 4 .20 0 0</td>
</tr>
<tr>
<td>AMT038D</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>AMT038H</td>
<td>ASIDname ASID %Bsy</td>
</tr>
<tr>
<td>AMT038I</td>
<td>AAON73 012A 40</td>
</tr>
<tr>
<td>AMT038I</td>
<td>AAOD72B 00A6 60</td>
</tr>
</tbody>
</table>

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, ********* 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 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**

![Diagram of MLCU](image)

The parameters are defined as follows:

- `lcu` logical control unit
- `lcu1-lcu2` range of logical control units; `lcu1` is the lowest-numbered logical control unit, and `lcu2` is the highest-numbered logical control unit
- `time` specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 1,200 seconds

**Example**

To monitor all LCUs for 15 seconds, type

```
mlcu
```
### 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.

<table>
<thead>
<tr>
<th>AMT061I LCU 023: CONTENTION RATE 0005 ALL PATHS BUSY 0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMT062I LCU 023: DELAY Q LENGTH 0000 CHPID TAKEN 0000 % CU BUSY 0000</td>
</tr>
</tbody>
</table>
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

The parameters are defined as follows:

- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information
- **seconds**: number of seconds that the address space is to be monitored; the default is 10 seconds, and the maximum is 1,200 seconds

Example

To monitor address space XTSTQPAS for 10 seconds, type

```
monitor,xtstqpas
```

<table>
<thead>
<tr>
<th>AMTJ22I Statistics being gathered for XTSTQPAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] [2] [3] [4]</td>
</tr>
<tr>
<td>AMTJ21I STC07754 XTSTQPAS XTSTQPAS, Service Class=STCNRM</td>
</tr>
<tr>
<td>[5] [6] [7] [8]</td>
</tr>
<tr>
<td>AMTJ23I Life of step CPU 305.91 Excp 529156 Pages 8410 SUs 6M</td>
</tr>
<tr>
<td>AMTJ24I 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
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**

```
MPATH [chpid] [chpid1-chpid2] [time]
```

The parameters are defined as follows:

- **chpid**: channel path identifier
- **chpid1-chpid2**: range of channel path identifiers; **chpid1** is the lowest channel path identifier, and **chpid2** is the highest channel path identifier
- **time**: specifies the length of the sample period in seconds; the default sample period is 15 seconds

**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

```
mpath
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO21I CHANNEL PATH 0C WAS</td>
<td>4% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 12 WAS</td>
<td>1% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 13 WAS</td>
<td>1% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 1B WAS</td>
<td>13% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 1C WAS</td>
<td>7% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 38 WAS</td>
<td>14% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 39 WAS</td>
<td>19% BUSY</td>
</tr>
<tr>
<td>AMTO21I CHANNEL PATH 3C WAS</td>
<td>19% BUSY</td>
</tr>
</tbody>
</table>
Example 2

To display I/O activity for channel paths 12–14, type

`mpath 12-14`

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTO21I CHANNEL PATH 12 WAS 4% 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 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 38 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
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:

`asi` address space instance; see “Identifying address spaces” on page 17 for more information
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>AMTMS2I</td>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>AMTMS3I Ext LSQA/SWA</td>
<td>7F6C8000</td>
<td>7FFFFFFF</td>
<td>9,440K</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Uncommitted Ext-Pvt</td>
<td>11ABA000</td>
<td>7F6C7FFF</td>
<td>1,798,392K</td>
<td>---</td>
<td>139FFFFF</td>
</tr>
<tr>
<td>AMTMS3I Committed Ext-Pvt</td>
<td>11A00000</td>
<td>11A89FFF</td>
<td>552K</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Ext CSA</td>
<td>051D0000</td>
<td>119FFFFFF</td>
<td>204,992K</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Ext MLPA</td>
<td>051BA000</td>
<td>051CFFFF</td>
<td>88K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Ext FLPA</td>
<td>051B7000</td>
<td>051B9FFF</td>
<td>12K</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Ext PLPA</td>
<td>0271A000</td>
<td>0271BFFF</td>
<td>552K</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I Ext SQA</td>
<td>0189F000</td>
<td>0189F000</td>
<td>664K</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I NUC R/W</td>
<td>01000000</td>
<td>01000000</td>
<td>360K</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I V=R (IF ANY)</td>
<td>00005000</td>
<td>00005000</td>
<td>360K</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>AMTMS3I System</td>
<td>------</td>
<td>-----</td>
<td>-------</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
- identifies the source of the malfunction when an address space stops processing
- finds out which address space requests a particular volume

**Syntax**

```
MTP
```

**Example**

To identify pending mount requests, type

```
mtp
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTM22I</td>
<td>MOUNT PENDING FOR B90078 UNIT= 384 (3400-4) JOB= GJJ$RUN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

- **SYSTEM**: specify SYSTEM, or no parameter, to display all system-level Name/Token pairs.
- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information.
- **seq#**: previously displayed sequence number.

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>Seq #</th>
<th>Name (Character)</th>
<th>Name (Hexadecimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*ISPFSVCX</td>
<td>C9E2D7C6 E2E5C3E7 000004FC</td>
</tr>
<tr>
<td>2</td>
<td><em>XDC]Z1DQXDC-XNTT</em></td>
<td>E7C4C3BD E9F1C4D8 E7C4C360</td>
</tr>
<tr>
<td>3</td>
<td>*IEWBILO/MMIB</td>
<td>C9C5E6C2 E7C9D3D6 61D4D4C9</td>
</tr>
<tr>
<td>4</td>
<td>*SYSPROG 3.7</td>
<td>E2E8E2D7 D9D6C740 F34BF740</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Seq #</th>
<th>Name (Character)</th>
<th>Name (Hexadecimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>*IRXTOKEN ?0</td>
<td>C9D9E7E3 D6D2C5D5 006FF070 2F309C90</td>
</tr>
<tr>
<td>6</td>
<td>*IRXTOKEN _ \</td>
<td>C9D9E7E3 D6D2C5D5 006DE1E0</td>
</tr>
<tr>
<td>7</td>
<td><em>SYSPROG 3.5 NTP1</em></td>
<td>E2E8E2D7 D9D6C740 F34BF540</td>
</tr>
<tr>
<td>8</td>
<td><em>SYSPROG 3.5 NTP2</em></td>
<td>E2E8E2D7 D9D6C740 F34BF540</td>
</tr>
</tbody>
</table>

**NOTE**

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>AMTNT5I Level=AdrsSpace BMVDWP3 (ASID 013F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTNT6I Name       *IEWBXIL0/MMIB      * C9C5E6C2 E7C9D3D6 61D4D4C9 C2404040</td>
</tr>
<tr>
<td>AMTNT7I Token       ** {            * 7F58FFC0 00000000 00000000 00000000</td>
</tr>
<tr>
<td>AMTNT0I</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
```
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.

### Syntax

The parameters are defined as follows:

- **StrExp**: 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 “C” 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.

  **Note**: 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 309.

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.
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:

```
AMTVY2I Found at: 00000000_00000010= MOBJ0005+1004
AMTVY3I 00000000_
AMTVY4I 41C01000 02D71000 E6D7C8C7 00000000 41C00000 *....WPHG.....(*...*
AMTVY4I 41C01010 00000000 41C02000 7F020D68 0202FFFF *.....(*............
AMTVY4I 41C01020 FF000000 FFFFF000 00000000 FFF00000 *.....0....0......0...
AMTVY4I 41C01030 7BB61668 7BB61668 00000000 00000000 *#...#................*
AMTVY4I 41C01040 00000000 00000000 80000000 00000000 *....................
AMTVY4I 41C01050 00000000 00000005 4001C000 00000000 *.........(......*
AMTVY4I 41C01060 80000000 00000000 00000000 00000000 *....................
AMTVY4I 41C01070 00000000 00000000 00000000 00000000 *....................
```
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

\[ \text{OFIND,MOBJ0005+10(8), MOBJ0005+18, BNDRY(8)} \]
The OKSWAP (OKS) service reverses the effects of the DONTSWAP service, making an address space swappable.

**Syntax**

```
OKSWAP  asi
```

The parameter is defined as follows:

*asi*  
address space instance; see “Identifying address spaces” on page 17 for more information

**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>[1]</th>
<th>AMTS61I TSU06701 INVENTRY IKJDWP1 Service Class=TSONRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>AMTS64I BMVDWP4 NONSWAPPABILITY COUNT REDUCED ( 1)</td>
</tr>
<tr>
<td>[3]</td>
<td>AMT001A SYSPROG</td>
</tr>
</tbody>
</table>
To make address space INVENTORY swappable, type

```plaintext
okswap, INVENTORY
```

### 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:

@ specifies your own address space
asi address space instance; see “Identifying address spaces” on page 17 for more information
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)`

<table>
<thead>
<tr>
<th>AMTVX0I</th>
<th>MOBJ</th>
<th>Origin</th>
<th>Size</th>
<th>Guard</th>
<th>Key</th>
<th>Owner</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVX1I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Created</td>
</tr>
<tr>
<td>AMTVX3I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Created</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Created</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Created</td>
</tr>
</tbody>
</table>

To display a list of the memory objects in job name MVQAZPAS and the user token for each object, type

`olist,mvqazpas,token`

<table>
<thead>
<tr>
<th>AMTVX0I</th>
<th>MOBJ</th>
<th>Origin</th>
<th>Size</th>
<th>Guard</th>
<th>Key</th>
<th>Owner</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVX2I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Token</td>
</tr>
<tr>
<td>AMTVX3I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Token</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Token</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
<td>Token</td>
</tr>
</tbody>
</table>

To display detailed information for a specific memory object, type

`ol,mobj0002`

| AMTVX5I | MOBJ0002 Owned by TCB 6BB9B0 in MVQAZPAS.P82A (X00AB) GL=HIGH |
| AMTVX6I | Store: Bgn 00000001_08000000, End 00000001_0FDFFFFF, Size= 126M |
| AMTVX7I | Guard: Bgn 00000001_0FE00000, End 00000001_0FFFFFFF, Size= 2M |
| AMTVX8I | Key 8, Fetch Protected |
| AMTVX9I | Alloc at 10:19:01.08 on 10/04/2004, Token=00000000_18460400 |
NOTE
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

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTVX2I</td>
<td>MOBJ</td>
<td>Origin</td>
<td>Size</td>
<td>Guard</td>
<td>Key</td>
<td>Owner</td>
</tr>
<tr>
<td>AMTVX3I</td>
<td>----</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>--</td>
<td>-------</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>0002 00000003_000000000_000000000_000000000</td>
<td>4066M</td>
<td>2078M&lt;</td>
<td>0p</td>
<td>7FD230 23:05</td>
<td>3/25/2005</td>
</tr>
<tr>
<td>AMTVX4I</td>
<td>0003 00000004_000000000_000000000_000000000</td>
<td>2029M</td>
<td>0M</td>
<td>0p</td>
<td>7FD230 23:05</td>
<td>3/25/2005</td>
</tr>
</tbody>
</table>

Legend:

1. a sequential number assigned by the OLIST service

The OLIST service creates a symbolic definition for each memory object, in the form MOBJnnnn. Symbols can be used with the DUMP, EQUATE, and ZAP services. For more details, see “Symbolic references” on page 316.

NOTE
The DUMP service can also generate symbol names in the form MOBJnnnn.

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>AMTC1H1 User defined symbols are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1GI MOBJ0001 00000001_00100000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0002 00000001_00A00000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0003 00000001_01500000</td>
</tr>
<tr>
<td>AMTC1GI MOBJ0004 00000001_01E00000</td>
</tr>
</tbody>
</table>

You can subsequently use these symbolic names to display the storage; for example:

du,mobj0001+60!

<table>
<thead>
<tr>
<th>AMTC1XI MOBJ0001+8000</th>
<th>Size= 127M, Guard= 1M</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTC1ZI 00000001_00008000 KEY= 8P BMVDWP4 (ASID 202)</td>
<td></td>
</tr>
<tr>
<td>AMTC12I 00008000 C1C2C3C4 40404040 D4D6C2D1 4EF6F040 *ABCD MOBJ+60 *</td>
<td></td>
</tr>
<tr>
<td>AMTC12I 00008010 606E40D4 D6C2D14E F8F0F0F0 40404040 *-&gt; MOBJ+8000 *</td>
<td></td>
</tr>
<tr>
<td>AMTC12I 00008020 00000000 00000000 00000000 00000000 <em>................</em></td>
<td></td>
</tr>
<tr>
<td>AMTC12I 00008030 00000000 00000000 00000000 00000000 <em>................</em></td>
<td></td>
</tr>
</tbody>
</table>
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**

```plaintext
paging
```

**Example**

To display paging information, type

```
paging
```

<table>
<thead>
<tr>
<th></th>
<th>TOTAL PAGING RATE</th>
<th>PAGES/SEC FOR LAST 65.25 SECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTP11</td>
<td>26.75 PAGES/SEC</td>
<td>[1]</td>
</tr>
<tr>
<td>AMTP12</td>
<td>10.05 PAGES/SEC</td>
<td>[2]</td>
</tr>
<tr>
<td>AMTP13</td>
<td>3.60 PAGES/SEC</td>
<td>[3]</td>
</tr>
<tr>
<td>AMTP14</td>
<td>9.75 PAGES/SEC</td>
<td>[4]</td>
</tr>
<tr>
<td>AMTP15</td>
<td>3.79 PAGES/SEC</td>
<td>[5]</td>
</tr>
<tr>
<td>AMTP16</td>
<td>6.44 PAGES/SEC</td>
<td>[6]</td>
</tr>
<tr>
<td>AMTP17</td>
<td>2.50 PAGES/SEC</td>
<td>[7]</td>
</tr>
</tbody>
</table>

**Legend:**

1. total system paging rate over the last data collection interval

   Total system paging is the sum of VIO paging operations (in and out), swapping operations, and all other paging operations (excluding CSA paging and page reclains).

2. duration of the last data collection interval

3. paging rate caused by virtual I/O

4. paging rate caused by swapping
5. paging rate in the system link-pack area (LPA)
6. paging rate in the common service area (CSA)
7. page reclaim rate

**Usage note**

The period for which paging data is reported can vary, 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:

*member* member in SYS1.PARMLIB

**Example**

To list the contents of SYS1.PARMLIB member IEASYS00, type

`parmlist ieasys00`

```
AMTCF1I  APF=00,
AMTCF1I  CMD=00,
```
AMTCF1I CSA=3072,
AMTCF1I CVIO,
AMTCF1I FIX=00,
AMTCF1I ICS=00,
AMTCF1I IOS=00,
AMTCF1I IPS=00,
AMTCF1I LNK=00,
AMTCF1I MLPA=(00),
AMTCF1I LOGCLS=L,
AMTCF1I LOGLMT=050000,
AMTCF1I MAXUSER=50,
AMTCF1I OPT=00,
AMTCF1I PAGE=(SYS1.PAGEA,
AMTCF1I    SYS1.LOCALA,
AMTCF1I    SYS1.LOCALB,
AMTCF1I    SYS1.LOCALC,
AMTCF1I    SYS1.LOCALD),
AMTCF1I PAGNUM=(4,3),
AMTCF1I REAL=192,
AMTCF1I RSU=2,
AMTCF1I SMF=00,
AMTCF1I SQA=8,
AMTCF1I SWAP=(SYS1.SWAPA
AMTCF1I    SYS1.SWAPB),
AMTCF1I VAL=00,
AMTCF1I VRREGN=192.
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 $$INEXCL, $$RPTEXT, $XENQ, or $$XRS.

**Syntax**

```
PBUILD member dsn
```

The parameters are defined as follows:

- **member**
  
  the member to be processed ($$INEXCL, $$RPJOBS, $$RPTEXT, $XENQ, or $$XRES)

- **dsn**
  
  *(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 SYSPROG Services was initialized.

**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
```

```
AMTPBII $$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:

- **LIST** displays all of the current PF key definitions
- **nn** indicates the number of the PF key whose definition you want to change
- **command** indicates the SYSPROG command to be invoked by the particular program function key

Example

To display the current PF key definitions, type

```
pfk list
```

```
AMTP811 TR
AMTP811 PA
AMTP811 01 ENQUEUES
AMTP811 02 RESERVES
AMTP811 03 END
AMTP811 04 CSA,MAP
AMTP811 05 MTP
AMTP811 06 PAGING
AMTP811 07 STATUS
AMTP811 08 REPLIES
AMTP811 09 SRM
AMTP811 10 CPU,5
AMTP811 11 CMD,||
AMTP811 12 IO
AMTP811 13 ENQUEUES
AMTP811 14 RESERVES
AMTP811 15 END
AMTP811 16 CSA,MAP
```
PFKEY

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'`

**NOTE**

Commands containing commas or embedded blanks must be enclosed in quotation marks.
The PIO service monitors I/O interrupt activity and determines the distribution of I/O interrupts among processors.

**Syntax**

```
PIO `time`
```

The parameter is defined as follows:

- `time` specifies the length of the sample period in seconds; the default sample period is 15 seconds, and the maximum period is 999 seconds.

**Example**

To display I/O interrupt activity, type

```
pio
```

<table>
<thead>
<tr>
<th>AMT070I</th>
<th>I/O Interrupt data being gathered for 15 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMT073I</td>
<td>Processor I/O Interrupt Activity Report</td>
</tr>
<tr>
<td></td>
<td>[1] [2] [3]</td>
</tr>
<tr>
<td>AMT074I</td>
<td>CPU 00 processed 676 interrupts (0% were through TPI)</td>
</tr>
<tr>
<td>AMT074I</td>
<td>CPU 10 processed 1,116 interrupts (1% were through TPI)</td>
</tr>
</tbody>
</table>

**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.
The PLIST (PLI) service displays the active exclusion list for $$INEXCL, $$RPJOBS, $$RPTEXT, $$XENQ, or $$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 parameters are defined as follows:

- `member`: the member to be processed ($$INEXCL, $$RPJOBS, $$RPTEXT, or $$XRES)

**Example**

To display the active exclusion list $$INEXCL.

```
plist, $$inexcl
```

```
AMTPL1I Exclusion list $$INEXCL created by BMVDWP3
AMTPL2I Monday, May 21, 2012; 1:35:03.69 PM Local
AMTPLEI Sampler Excluded Jobnames
AMTPLFI
AMTPL3I AIO ---JES2
AMTPL3I AIOR---JES2
AMTPL3I APG ---MTACDMDB MTACDMDB MTA$DBCD
```
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 in the description of the TCB service on page 411.

### Syntax

```
POST expression asi [code]
```

The parameter is defined as follows:

- **expression** is an expression that resolves to the ECB address, which can (and generally is) a symbol created by the TCB service.
- **asi** is required if the expression that is specified as the first parameter does not imply an address space.

The symbols that are generated by the TCB service imply the address space containing the TCB. Typically, this parameter is not needed.
- **code** is an optional post code for which zero is used if omitted.

### Example

Assume that the following AMTT76I messages were displayed for a TCB in address space, MYSPACE:

```
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,006C1D28,16,myspace
```
The PPT service displays and updates the program properties table (PPT).

**Syntax**

```plaintext
PPT [-name name [-prefix prefix*]]
```

**Description**

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.

<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>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>NOSPREF</td>
<td>NOSPREF or NO2P is the IBM default value; SPREF or 2P is the opposite value.</td>
</tr>
<tr>
<td>NO2P</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>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>

**Enter**

- `ppt` display all entries in the PPT
- `ppt,prefix*` display all PPT entries that begin with the `prefix`
- `ppt,name` display the PPT entry for a program `name`
**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>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I ISTINMO1</td>
<td>Yes</td>
<td>Yes</td>
<td>6</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I IKTCAS00</td>
<td>Yes</td>
<td>No</td>
<td>6</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I AHLGTF</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I HHLGTF</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I IHGTF</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I IEFIIC</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IEEBB60</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IEEVMNT2</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IASXWRO0</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I CSVVFCRE</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I HASJES20</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I DFSMVRCO</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I IATINTK</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I DXRRLM00</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I APSPPIEP</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I APKCSIEP</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I IATINTKF</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I DSNYASC3</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I DSNUTLBD</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IEAVTDSV</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IFASMF</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I CSVLLCRE</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I AVFMNBLD</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
</tr>
<tr>
<td>AMTPP4I ERBMFMC</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I ERB3GMFC</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>AMTPP4I IGG0CLX0</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>SPREF</td>
</tr>
<tr>
<td>AMTPP4I IGDSSI01</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

For example: PPT, *name*, NOSWAP,CANCEL would add the NOSWAP attribute, remove the NOCANCEL attribute, and leave all others unchanged.

```
ppt,name,NOHUSI
```

modify the PPT entry for the program *name* to ignore IEFUSI region controls.
Example 2

To display all PPT entries beginning with IEDQ, type

```
ppt,iedq*
```

<table>
<thead>
<tr>
<th>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>AMTPP1I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTPP2I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 3

To reset all attributes for program name DWPTEST, if it exists in the table, type

```
ppt,dwpertest,reset
```

| Add/Update successful.
<table>
<thead>
<tr>
<th>AMTPP0I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Cancel</th>
<th>Swap</th>
<th>Key</th>
<th>Priv</th>
<th>Task</th>
<th>Integ</th>
<th>Pswd</th>
<th>HUSI</th>
<th>Storage</th>
<th>Pref</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPP3I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTPP4I</td>
<td>DWPTEST</td>
<td>No</td>
<td>Yes</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>NOPREF</td>
<td></td>
</tr>
<tr>
<td>AMTPP0I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPRG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 4

To set two attributes in the entry for DWPTEST, if it exists in the table, type

```
ppt,dwpertest,nocan,noswap
```

| Add/Update successful.
<table>
<thead>
<tr>
<th>AMTPP0I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Cancel</th>
<th>Swap</th>
<th>Key</th>
<th>Priv</th>
<th>Task</th>
<th>Integ</th>
<th>Pswd</th>
<th>HUSI</th>
<th>Storage</th>
<th>Pref</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPP3I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTPP4I</td>
<td>DWPTEST</td>
<td>Yes</td>
<td>Yes</td>
<td>8</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTPP0I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPRG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE

If DWPTEST does not currently exist, it is created.
Example 5

To change the key for entry DWPTEST, if it exists, type

`ppt,dwptest,key(4)`

**NOTE**

If DWPTEST does not currently exist, it is created.
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

The parameters are defined as follows:

@ your own address space (default)
asi address space instance; see “Identifying address spaces” on page 17 for more information
LSQA detailed report of allocations in the Local Systems Queue Area (LSQA), which consists of subpools 203-205, 213-215, 223-225, and 253-255
spid three-digit decimal subpool identifier in the range between 0 and 255
TCB summary of the storage allocated by all tasks in the address space
address address of the TCB for which you want to display a detailed report of storage allocations
HIGH summary of subpools located in private high storage
LOW summary of subpools located in private low storage
REGION
PVT summary of subpools 0–127, 129–132, 229, 230, 236, 237, 251, and 252

SWA detailed report of allocations in the Scheduler Work Area (SWA), which consists of subpools 236 and 237

REGION regions for the specified address space

Example 1

To determine LSQA usage for address space INVENTRY, type

private inventry

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPRBI 255,ALLOC</td>
<td>240K [3]</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>7K [4]</td>
</tr>
<tr>
<td>AMTPRBI 205,ALLOC</td>
<td>4K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>0K</td>
</tr>
<tr>
<td>AMTPRBI 215,ALLOC</td>
<td>284K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>3K</td>
</tr>
<tr>
<td>AMTPRBI 225,ALLOC</td>
<td>16K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>8K</td>
</tr>
<tr>
<td>AMTPRBI 255,ALLOC</td>
<td>8648K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>1K</td>
</tr>
<tr>
<td>AMTPRTI TOT,ALLOC</td>
<td>9192K [5]</td>
</tr>
<tr>
<td>AMTPRTI</td>
<td>19K [6]</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
Example 2

To summarize storage allocations by subpool and key for all TCBs belonging to address space INVENTORY, type

```
private inventry tcb
```

<table>
<thead>
<tr>
<th>AMTPRBI</th>
<th>SUMMARY OF PRIVATE STORAGE FOR INVENTORY BY TCB, SUBPOOL AND KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPRBI</td>
<td>[1] [2] [3]</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td># 1 TCB=7FE158, RB=007FDC58, PGM=IEAVAR00</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRHI</td>
<td>SP KEY ABOVE BELOW OWNER</td>
</tr>
<tr>
<td>AMTPRLI</td>
<td>ALLOC FFS ALLOC FFS</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>229 0 24K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>230 0 60K 6K 8K 2K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>TOTAL 84K 6K 8K 2K [11]</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td># 2 TCB=7FF7C0, RB=007F3D60, PGM=IEFSD060</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRHI</td>
<td>SP KEY ABOVE BELOW OWNER</td>
</tr>
<tr>
<td>AMTPRLI</td>
<td>ALLOC FFS ALLOC FFS</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>0 0 4K 2K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>229 0 4K 3K 36K 6K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>230 0 468K 21K 56K 15K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>236 0 628K 33K 104K 16K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>237 0 636K 33K 124K 21K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>252 0 636K 33K 128K 24K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>TOTAL 2372K 123K 452K 84K</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td># 3 TCB=7FF228, RB=007FF128, PGM=IKJEFT01</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRHI</td>
<td>SP KEY ABOVE BELOW OWNER</td>
</tr>
<tr>
<td>AMTPRLI</td>
<td>ALLOC FFS ALLOC FFS</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>0 0 4K 2K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>1 0 12K 5K 8K 3K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>2 0 4K 3K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>78 0 124K 7K 84K 12K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>230 0 44K 7K 56K 6K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>236 0 204K 19K 104K 7K SHR</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>251 0 300K 47K 280K 34K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>252 0 500K 52K 480K 43K OWN</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>==============================================================</td>
</tr>
<tr>
<td>AMTPRVI</td>
<td>TOTAL 1184K 137K 1020K 110K</td>
</tr>
</tbody>
</table>
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 INVENTORY, type

`private inventry region`

<table>
<thead>
<tr>
<th>AMTPR2I</th>
<th>LIST OF REGIONS FOR PRIVATE ADDRESS SPACE INVENTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPR2I</td>
<td>REGION ADDRESS LENGTH</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>RCT 00001000 00004000</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>V=V 00005000 007FB000</td>
</tr>
<tr>
<td>AMTPRBI</td>
<td>E V=V 05800000 7A800000</td>
</tr>
</tbody>
</table>
The PROGRESS (PRO) service displays status and current step information for the specified address space.

Currently, two versions of the PROGRESS service are supported. This section documents the new version. The new version is the default unless SYSPROG Services is operating under the control of AutoOPERATOR, in which case the PROGRESS version distributed in SYSPROG Services 3.6 (as documented in Appendix B, “Previous version of the Progress service”) is the default. You can use the version parameter to select your preferred PROGRESS version.

**Syntax**

```
PROGRESS  asi  version
```

Where:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asi</td>
<td>&quot;address space instance&quot;; see “Identifying address spaces” on page 17 for more information.</td>
</tr>
<tr>
<td>version</td>
<td>The version number. Specify “3” for the new version and “2” for the prior version. “2” is the default when SYSPROG Services is operating under AutoOPERATOR™, otherwise the default is “3”.</td>
</tr>
</tbody>
</table>

**Example 1**

To display the progress of address space BMVDWP3 (which happens to be a TSO session), type

```
pro, bmvdwp4
```

```
AMTPG0I  TSU03895  BMVDWP4          Service Class= TSONRM
[1]
AMTPGTI  Swaped out due to: Long wait [28]
AMTPGRI  Reusable ASCB=NO          [2]
AMTPG3I  MSGCLASS= R               MSGLEVEL=(1,1)  [3]
AMTPG7I  Program=IKJEFT01          Pgmrm Name=     [4]
AMTPG1I  Job Start Date=  24 May 2012  2012.145 [5]
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPG2I</td>
<td>Job Start Time</td>
<td>8:47:26.960</td>
</tr>
<tr>
<td>AMTPG4I</td>
<td>Step Start Time</td>
<td>8:47:26.960</td>
</tr>
<tr>
<td>AMTPG5I</td>
<td>Step TCB Time</td>
<td>0:00:00.716</td>
</tr>
<tr>
<td>AMTPGMI</td>
<td>Step TCB Limit</td>
<td>23:59:00.796</td>
</tr>
<tr>
<td>AMTPGNI</td>
<td>zAAP CPU time</td>
<td>0:00:00.000</td>
</tr>
<tr>
<td>AMTPGNI</td>
<td>zIIP CPU time</td>
<td>0:00:00.000</td>
</tr>
<tr>
<td>AMTPGUI</td>
<td>Step Ph tm Time</td>
<td>0:00:00.002</td>
</tr>
<tr>
<td>AMTPGFI</td>
<td>Step ASST Time</td>
<td>0:00:00.000</td>
</tr>
<tr>
<td>AMTPGSI</td>
<td>Region Limit set by JCL</td>
<td></td>
</tr>
<tr>
<td>AMTPGSI</td>
<td>Region Requested</td>
<td>64Mi</td>
</tr>
<tr>
<td>AMTPGSI</td>
<td>Rgn Bot Limit Below</td>
<td>7,144Ki</td>
</tr>
<tr>
<td>AMTPGCI</td>
<td>Cur Bot Rgn Below</td>
<td>364Ki</td>
</tr>
<tr>
<td>AMTPGDI</td>
<td>Max Bot Rgn Below</td>
<td>456Ki</td>
</tr>
<tr>
<td>AMTPGV1I</td>
<td>Cur Top Rgn Below</td>
<td>504Ki</td>
</tr>
<tr>
<td>AMTPGWI</td>
<td>Max Top Rgn Below</td>
<td>556Ki</td>
</tr>
<tr>
<td>AMTPGJI</td>
<td>Common Storage</td>
<td>---CSA----</td>
</tr>
<tr>
<td>AMTPGKI</td>
<td>Below</td>
<td>136</td>
</tr>
<tr>
<td>AMTPGLI</td>
<td>Above</td>
<td>3,656</td>
</tr>
<tr>
<td>AMTPGGI</td>
<td>MEMLIMIT (LVMO)</td>
<td>20Gi</td>
</tr>
<tr>
<td>AMTPGHI</td>
<td>Current</td>
<td>8,192Ki</td>
</tr>
<tr>
<td>AMTPGII</td>
<td>High-Water Mark</td>
<td>9,216Ki</td>
</tr>
<tr>
<td>AMTPGEI</td>
<td>LNKLST Set Name</td>
<td>IPL</td>
</tr>
</tbody>
</table>

Job Class = *
Example 2

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.

```
pro.bmvdwp4
  AMTPG0I TSU03895 BMVDWP4       Service Class= TSONRM
[1]
  AMTPGQI Address space BMVDWP4 is quiesced
[27]
  AMTPGJI Common Storage       ---CSA----   ---SQA----
[20]
  AMTPGKI Below                 136            96
[21]
  AMTPGLI Above                 3,656          368
[22]
  AMTPGGI MEMLIMIT (LVMO)=      20Gi           Set by= JCL
[23]
  AMTPGHI Current=             8,192Ki         Hidden= 0
[24]
  AMTPGII High-Water Mark=     9,216Ki          # Objects= 8
[25]
  AMTPGEI LNKLST Set Name= IPL
[26]
```

**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.

23. maximum amount of above-the-bar private storage that the address space may allocate and how that limit was established.
24. current amount of above-the-bar private storage allocated by the address space and the amount that is hidden

25. high-water mark (maximum amount) of above-the-bar-storage allocated by this address space

26. name of the address space's LNKLST set

27. message that is displayed when the address space has been quiesced

28. message that indicates the swap reason displayed when the address space is swapped out
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**

```
PRSM [10] time ALL
```

The parameters are defined as follows:

- **time**: length of the sample period in seconds; the default sample period is 10 seconds, and the maximum period is 300 seconds.
- **ALL**: includes inactive as well as active partitions in the display.
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 | [1] |
| AMTL11I | Partitions: 24; CPUs: CP 18, ICF 4 |

<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>AMTL3I</td>
<td>CPU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL4I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL5I</td>
<td>CP</td>
<td>2</td>
<td>90</td>
<td>3.7</td>
<td>21.3</td>
<td>2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL6I</td>
<td>ICF</td>
<td>1</td>
<td>400</td>
<td>50.0</td>
<td>97.9</td>
<td>24.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL7I</td>
<td>ICF</td>
<td>1</td>
<td>-DED-</td>
<td>100.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL8I</td>
<td>ICF</td>
<td>1</td>
<td>100</td>
<td>12.5</td>
<td>90.8</td>
<td>22.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL9I</td>
<td>ICF</td>
<td>1</td>
<td>300</td>
<td>37.5</td>
<td>90.9</td>
<td>22.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL10I</td>
<td>CP</td>
<td>2</td>
<td>140</td>
<td>5.8</td>
<td>10.2</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL11I</td>
<td>CP</td>
<td>3</td>
<td>200</td>
<td>8.2</td>
<td>74.6</td>
<td>12.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL12I</td>
<td>CP</td>
<td>1</td>
<td>42</td>
<td>1.7</td>
<td>9.2</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL13I</td>
<td>CP</td>
<td>1</td>
<td>53</td>
<td>2.1</td>
<td>7.4</td>
<td>.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL14I</td>
<td>CP</td>
<td>2</td>
<td>114</td>
<td>4.7</td>
<td>9.5</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL15I</td>
<td>CP</td>
<td>2</td>
<td>120</td>
<td>4.9</td>
<td>12.4</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL16I</td>
<td>CP</td>
<td>2</td>
<td>140</td>
<td>5.8</td>
<td>18.4</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL17I</td>
<td>CP</td>
<td>3</td>
<td>200</td>
<td>8.2</td>
<td>63.6</td>
<td>10.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL18I</td>
<td>CP</td>
<td>2</td>
<td>260</td>
<td>10.7</td>
<td>78.2</td>
<td>8.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL19I</td>
<td>CP</td>
<td>2</td>
<td>140</td>
<td>5.8</td>
<td>36.8</td>
<td>4.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL20I</td>
<td>CP</td>
<td>2</td>
<td>-DED-</td>
<td>100.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL21I</td>
<td>CP</td>
<td>2</td>
<td>140</td>
<td>5.8</td>
<td>76.3</td>
<td>8.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL22I</td>
<td>CP</td>
<td>2</td>
<td>122</td>
<td>5.0</td>
<td>16.7</td>
<td>1.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL23I</td>
<td>CP</td>
<td>2</td>
<td>114</td>
<td>4.7</td>
<td>47.9</td>
<td>5.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL24I</td>
<td>CP</td>
<td>2</td>
<td>130</td>
<td>5.3</td>
<td>6.7</td>
<td>.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL25I</td>
<td>CP</td>
<td>2</td>
<td>115</td>
<td>4.7</td>
<td>16.4</td>
<td>1.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL26I</td>
<td>CP</td>
<td>2</td>
<td>120</td>
<td>4.9</td>
<td>24.1</td>
<td>2.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL27I</td>
<td>CP</td>
<td>2</td>
<td>140</td>
<td>5.8</td>
<td>26.5</td>
<td>2.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTL28I</td>
<td>CP</td>
<td>1</td>
<td>30</td>
<td>1.2</td>
<td>.0</td>
<td>.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AMTL01I Processor Type: CP [14]
AMTL02I Plex busy: 68.57% [15]
AMTL03I Plex overhead: 11.10% [16]
AMTL04I Plex wait: 20.33% [17]
AMTL05I

AMTL01I Processor Type: ICF [14]
AMTL02I Plex busy: 69.90% [15]
AMTL03I Plex overhead: 4.60% [16]
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.

9. 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.

10. weight that the dispatcher for PR/SM will give to this logical partition

11. partition’s weight divided by the sum of the weights for all partitions in the plex

12. average logical CPU use that this logical partition is making of the processor complex

13. average physical CPU use that this logical partition is making of the processor complex

14. physical processor utilization across the plex, grouped by type of processor
15. percentage of time that this type of processor was gainfully employed

16. percentage of this type of processor’s time that was used to determine how the processors should be dispatched to the various LPARs

17. percentage of time that this type of processor was not used

18. 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.

19. processor ID; one line for each physical processor that is assigned to the LPAR

20. 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
The PSTORAGE (PST) service provides the ability to determine

- amount of private storage allocated by the various tasks within an address space
- private storage allocated to each subpool associated with a task
- 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**

```
PSTORAGE asi symbol spn
```

The parameters are defined as follows:

- `asi` address space instance; see “Identifying address spaces” on page 17 for more information
- `symbol` symbolic name in the form @TCBnnn

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.
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

```
ps,grs
```

<table>
<thead>
<tr>
<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>@TCB001</td>
<td>7FE0A8</td>
<td>IEAVAR00</td>
<td>126,976</td>
<td>47,736</td>
<td>79,240</td>
</tr>
<tr>
<td>@TCB002</td>
<td>7FFBF8</td>
<td>IEEPRWI2</td>
<td>237,387,776</td>
<td>237,296,632</td>
<td>91,144</td>
</tr>
<tr>
<td>@TCB003</td>
<td>7CEE88</td>
<td>ISGWDRVR</td>
<td>65,536</td>
<td>65,536</td>
<td>0</td>
</tr>
<tr>
<td>@TCB004</td>
<td>7FF6F8</td>
<td>ISQDRVR</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>@TCB005</td>
<td>7FF918</td>
<td>ISCMCDRT</td>
<td>8,192</td>
<td>8,192</td>
<td>0</td>
</tr>
<tr>
<td>@TCB006</td>
<td>7F52E8</td>
<td>ISGCCNST</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>@TCB007</td>
<td>7F5480</td>
<td>ISGTSSMT</td>
<td>135,168</td>
<td>131,184</td>
<td>3,984</td>
</tr>
<tr>
<td>@TCB008</td>
<td>7FFE88</td>
<td>IEAVTSDT</td>
<td>69,632</td>
<td>59,272</td>
<td>10,360</td>
</tr>
<tr>
<td><strong>Total</strong></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

```
ps,@tcb002
```

<table>
<thead>
<tr>
<th>SPN</th>
<th>Allocated</th>
<th>Obtained</th>
<th>Frag Free</th>
<th>% Frag</th>
</tr>
</thead>
<tbody>
<tr>
<td>O/S</td>
<td>0</td>
<td>4,096</td>
<td>472</td>
<td>3.624</td>
</tr>
<tr>
<td>Own</td>
<td>127</td>
<td>236,171,264</td>
<td>236,158,336</td>
<td>12,928</td>
</tr>
<tr>
<td>Own</td>
<td>229</td>
<td>724,992</td>
<td>701,520</td>
<td>23,472</td>
</tr>
<tr>
<td>Own</td>
<td>230</td>
<td>90,112</td>
<td>48,672</td>
<td>41,440</td>
</tr>
<tr>
<td>Own</td>
<td>252</td>
<td>397,312</td>
<td>387,632</td>
<td>9,680</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>237,387,776</td>
<td>237,296,632</td>
<td>91,144</td>
</tr>
</tbody>
</table>
**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

```
ps,127
```

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>GRS</th>
<th>IEEPRWI2 @TCB002 at 7FFBF8, SPN=127 Owned</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>Key</td>
<td>Allocated</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>---</td>
<td>----------</td>
</tr>
<tr>
<td>AMTPQ8I</td>
<td>0</td>
<td>236,171,264</td>
</tr>
<tr>
<td>AMTPQ9I</td>
<td>------</td>
<td>Below the Line Storage------</td>
</tr>
<tr>
<td>AMTPQ11</td>
<td>Total</td>
<td>236,171,264</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

```
ps,@tcb004
```

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>GRS</th>
<th>ISGQDRVR @TCB004 at 7FF6F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>SPN</td>
<td>Allocated</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>---</td>
<td>----------</td>
</tr>
<tr>
<td>AMTPQ8I</td>
<td>Shr</td>
<td>4,096</td>
</tr>
<tr>
<td>AMTPQ9I</td>
<td>Total</td>
<td>4,096</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

\texttt{ps,@tcb002,229}

<table>
<thead>
<tr>
<th>AMTPQ5I</th>
<th>GRS</th>
<th>IEEPRWI2</th>
<th>@TCB002 at 7FFBF8, SPN=229 Owned</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ6I</td>
<td>Key</td>
<td>Allocated</td>
<td>Obtained</td>
</tr>
<tr>
<td>AMTPQ7I</td>
<td>---</td>
<td>552,960</td>
<td>539,168</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMTPQKI</th>
<th>Below the Line Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTPQ7I</td>
<td>---</td>
</tr>
</tbody>
</table>

| AMTPQ7I | --- | 724,992 | 701,520 | 23,472 | 3.24 |

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

\texttt{ps}

<table>
<thead>
<tr>
<th>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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>AMTPQWI</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


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 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.

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.

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 39.

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:

**RNAME=rname**

RNAME for the search

Specify a 1- to 64-character literal value (see the explanation of literal value in the section “Definitions” on page 354). 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 354.

**QNAME=qname**

identifies the QNAME value for the search

Specify a one- to eight-character literal value (see the explanation of literal value in the section “Definitions” on page 354). 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 354.

**JOBNAME=jobname**

(optional) when specified, limits the returned information to the specified address space

 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 17 for the various forms that can be used to identify the address space.

The JOBNAME parameter is mutually exclusive with the TASK= parameter.

**SCOPE=keyword**

defines the SCOPE of the resources that are to be returned

The following values are allowed: ANY, SYSTEM, SYSTEMS, and SYSPLEX. The default is ANY.

**TASK= symbolName**

(optional) specifies a symbol that identifies the desired task that was created by the TCB or PSTORAGE service

The TASK parameter is mutually exclusive with the JOB parameter.

**RTOKEN**

(optional) indicates that the 32-byte resource token is to be displayed for each listed resource

You can specify NO or YES. The default is NO

RTOKEN=YES is supported only on z/OS 1.9 and later.
### 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 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 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 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 quotation mark; therefore, it is considered to be a two-byte hexadecimal value. The case is ignored.</td>
</tr>
<tr>
<td>abcd(64)</td>
<td>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 18 for additional information.</td>
</tr>
</tbody>
</table>

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*`
Example 2

To display enqueue information for a specific data set, including the resource token, type

`que,vam3.gen.load`

---

**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

\texttt{que,task=@tcb2}

<table>
<thead>
<tr>
<th>AMTQU1I</th>
<th>Your search Located 152 resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQU2I</td>
<td>Scope=SYSTEM Owners= 2 No Waiters</td>
</tr>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=RXA36.VAM3.BBPLIB</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID Jobname ASID TCB Stat Typ Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD BMVDWP4 0140 @TCB2 Owns Shr 8:34:00 5/23/2012</td>
</tr>
<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=SYS0.CPWFILGA.V8ROMO.ISPMLIB</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID Jobname ASID TCB Stat Typ Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD BMVDWP4 0140 @TCB2 Owns Shr 8:34:00 5/23/2012</td>
</tr>
<tr>
<td>AMTQU2I</td>
<td>Scope=SYSTEM Owners= 9 No Waiters</td>
</tr>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=SYS1.IBMREX.SYS.SFANSKL</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID Jobname ASID TCB Stat Typ Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD BMVDWP4 0140 @TCB2 Owns Shr 8:34:00 5/23/2012</td>
</tr>
<tr>
<td>AMTQU2I</td>
<td>Scope=SYSTEM Owners= 15 No Waiters</td>
</tr>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=E0Y.PREP.SEOYPENU</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID Jobname ASID TCB Stat Typ Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD BMVDWP4 0140 @TCB2 Owns Shr 8:34:00 5/23/2012</td>
</tr>
<tr>
<td>AMTQU2I</td>
<td>Scope=SYSTEM Owners= 20 No Waiters</td>
</tr>
<tr>
<td>AMTQU3I</td>
<td>Q=SYSDSN</td>
</tr>
<tr>
<td>AMTQU4I</td>
<td>R=SYS1.SBLSPNL0</td>
</tr>
<tr>
<td>AMTQU5I</td>
<td>SYSID Jobname ASID TCB Stat Typ Time</td>
</tr>
<tr>
<td>AMTQU6I</td>
<td>SJSD BMVDWP4 0140 @TCB2 Owns Shr 8:34:00 5/23/2012</td>
</tr>
</tbody>
</table>
Example 4

To display ENQ conflicts, type

que

NOTE

The TCB service was used to assign symbolic names to the address space’s tasks.
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 ALL
```

The parameters are defined as follows:

- `systemName` one- to eight-character system name
- `ALL` displays replies from all systems

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 “$$RPJOBS” on page 37) or containing specific text (see “$$RPTEXT” on page 38).
Example 1

To display outstanding replies from the current system, type

rep

<table>
<thead>
<tr>
<th>AMTR101</th>
<th>AMTR16I</th>
<th>AMTR12I</th>
<th>AMTR17I</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td>SYSNAME</td>
<td>SJSC</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>[1]</td>
<td>[2]</td>
<td>[3]</td>
<td></td>
</tr>
<tr>
<td>AMTR16I</td>
<td>SYSNAME</td>
<td>SJSC</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>AMTR12I</td>
<td>SJSC</td>
<td>SJSC</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
</tr>
<tr>
<td>AMTR17I</td>
<td>SJSC</td>
<td>SJSC</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>AMTR16I</th>
<th>AMTR12I</th>
<th>AMTR17I</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSNAME</td>
<td>SJSC</td>
<td>SJSC</td>
</tr>
<tr>
<td>JOB ID</td>
<td>ENTERED - AWAITING PROGRAMMER SIGNON. REPLY</td>
<td></td>
</tr>
<tr>
<td>[1]</td>
<td>[2]</td>
<td>[3]</td>
</tr>
<tr>
<td>AMTR16I</td>
<td>STC03223</td>
<td>STC01830</td>
</tr>
<tr>
<td>AMTR12I</td>
<td>STC03223</td>
<td>STC01830</td>
</tr>
<tr>
<td>AMTR17I</td>
<td>STC03223</td>
<td>STC01830</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
The RESERVES (RES) service

- helps determine the cause of poor DASD performance or system lockouts
- 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 40.

**Syntax**

```
RESERVES
```

**Example**

To display reserve activity on DASD devices, type

```
reserves
```
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 (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. 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
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**

```plaintext
RSM

| MAP | ASID | NAME | FRAMES | FIXED | <16MB | LSQA | WSS | PERCENT |
```

The parameters are defined as follows:

- **MAP** provides a tabular display of the amount of real storage allocated to each address space
- **ASID** sorts the output in ascending order by address space identifier
- **NAME** sorts the output in alphabetical order by address space name or system area
- **FRAMES** 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
- **FIXED** 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
Example 1

To display real-storage allocation, type

```
rsm
```

<table>
<thead>
<tr>
<th>Address Space</th>
<th>Total Frames</th>
<th>Fixed Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTRS0I</td>
<td>Total</td>
<td>Fixed</td>
</tr>
<tr>
<td>AMTRS1I</td>
<td>Private frames: 222157</td>
<td>Fixed frames: 27295 [1] [2]</td>
</tr>
<tr>
<td>AMTRS4I</td>
<td>Free frames : 4713</td>
<td>Fixed frames : [7] [8]</td>
</tr>
<tr>
<td>AMTRS5I</td>
<td>Total online : 255706</td>
<td>Fixed frames : 38611 (&lt;16MB 693) [9] [10]</td>
</tr>
<tr>
<td>AMTRS7I</td>
<td>PFT frames : 4109</td>
<td>Fixed frames : [13] [14]</td>
</tr>
<tr>
<td>AMTRS8I</td>
<td><em>Unknown</em> : 6</td>
<td>Fixed frames : [15]</td>
</tr>
<tr>
<td>AMTRS9I</td>
<td>Total : 262144 (1024M)</td>
<td>Fixed frames : 40934 [16] [17]</td>
</tr>
<tr>
<td>AMT001A SYSPROG</td>
<td>Total : 262144 (1024M)</td>
<td>Fixed frames : 40934 [16] [17]</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)

<16MB 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

LSQA 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

WSS 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

PERCENT 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
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>AMTR50I</th>
<th>Total</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTR51I</td>
<td>Private Frames: 218716</td>
<td>29085</td>
</tr>
<tr>
<td>AMTR52I</td>
<td>Common Frames : 14476</td>
<td>630</td>
</tr>
<tr>
<td>AMTR53I</td>
<td>SQA Frames : 16243</td>
<td>16243</td>
</tr>
<tr>
<td>AMTR54I</td>
<td>Free Frames : 6262</td>
<td></td>
</tr>
<tr>
<td>AMTR55I</td>
<td>Total Online : 255697</td>
<td>45958 (&lt;16MB 737)</td>
</tr>
</tbody>
</table>

| AMTR56I | Nucleus : 2332 | 2332 |
| AMTR57I | PFT Frames : 4109 | |
| AMTR58I | *Unknown* : 6 | |
| AMTR59I | Total : 262144 ( 1024M) | 48290 |

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTR25I</td>
<td>NAME</td>
<td>ASID</td>
<td>FRAMES</td>
<td>FIXED</td>
<td>&lt; 16MB</td>
<td>LSQA</td>
<td>WSS</td>
</tr>
<tr>
<td>AMTR26I</td>
<td>* FREE *</td>
<td>****</td>
<td>6262</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>AMTR26I</td>
<td>SQA</td>
<td>****</td>
<td>16243</td>
<td>16243</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>AMTR26I</td>
<td>LPA</td>
<td>****</td>
<td>3840</td>
<td>125</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AMTR26I</td>
<td>CSA</td>
<td>****</td>
<td>10636</td>
<td>505</td>
<td>32</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>AMTR26I</td>
<td>NUCLEUS</td>
<td>****</td>
<td>2332</td>
<td>2332</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

```
----------------------- Sorted Data --------------------------
```

```
| AMTR26I | *MASTER* 0001 | 3241 | 3037 | 0 | 187 | 0 | 1 |
| AMTR26I | PCAUTH 0002 | 734 | 664 | 0 | 664 | 0 | 0 |
| AMTR26I | RASP 0003 | 360 | 355 | 0 | 43 | 0 | 0 |
| AMTR26I | TRACE 0004 | 151 | 105 | 0 | 104 | 0 | 0 |
| AMTR26I | DUMPSRV 0005 | 173 | 119 | 0 | 119 | 0 | 0 |
| AMTR26I | XCFAS 0006 | 9108 | 2008 | 0 | 375 | 0 | 3 |
| AMTR26I | GRS 0007 | 964 | 182 | 17 | 50 | 0 | 0 |
| AMTR26I | SMSPDSE 0008 | 1836 | 124 | 0 | 49 | 0 | 0 |
| AMTR26I | SMSVSAM 0009 | 2195 | 403 | 0 | 245 | 0 | 0 |
| AMTR26I | CONSOLE 000A | 3208 | 137 | 0 | 78 | 0 | 1 |
| AMTR26I | WLM 000B | 2161 | 136 | 0 | 78 | 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:

\[
\frac{\text{Total frames (pageable and fixed) allocated to the address space or system area}}{\text{RCEPOOL}} \times 100\%
\]

If the result of this calculation yields a result less than 1%, the field is replaced by dashes (--).

---

The Set Counter (SCT) service provides the ability to enable/disable and activate/deactivate one or more sampler sets.

**Syntax**

```
sct setname, action
```
The parameters are defined as follows:

<table>
<thead>
<tr>
<th>setname</th>
<th>is one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROBLEM-STATE</td>
<td></td>
</tr>
<tr>
<td>BASIC</td>
<td></td>
</tr>
<tr>
<td>EXTENDED</td>
<td></td>
</tr>
<tr>
<td>COPROCESSOR</td>
<td></td>
</tr>
<tr>
<td>CRYPTO-ACTIVITY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>action</th>
<th>is one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENABLE</td>
<td></td>
</tr>
<tr>
<td>DISABLE</td>
<td></td>
</tr>
<tr>
<td>ACTIVATE</td>
<td></td>
</tr>
<tr>
<td>DEACTIVATE</td>
<td></td>
</tr>
</tbody>
</table>

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 410 for more information.
Syntax

The parameters are defined as follows:

- **asi**: address space instance; see “Identifying address spaces” on page 17 for more information
- **title**: title that is recorded in the first line of the SVC dump and is displayed by IPC $S$

The title must be enclosed in single quotes if it contains embedded blanks.

- **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
- **NOLPA**: suppresses display of the active link-pack area modules and SVCs; these areas are dumped by default
- **NOLSQA**: suppresses display of the LSQA and ELSQA subpools; these subpools are dumped by default
- **NOPSA**: 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
Examples

To obtain an SVC dump of three address spaces, adding the title `DUMP1`, type

```
sdump,(xtstqcas,xtstqpas,(1)),’DUMP1’
```

- 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).

- The second parameter is required and is the title for the dump (in this case, `DUMP1`).

- No dump options were specified (third parameter position); therefore, the default dump options are used.

To obtain an SVC dump of the Master address space expressed as an ASID, type

```
sdump,(1),title,(nuc,nolpa,nocsa,nosqa)
```
The word *title* is provided as the title for the dump; it is not enclosed in single quotes because it does not contain any blanks or special characters.

The third parameter is a list of dump options, enclosed in parentheses. These options will be merged with, and take precedence over, the default options.

**Legend:**

1. The first message indicates that an SVC dump is being taken.

2. Prior to taking the dump, the SDUMP service obtains a token for the dump and displays the token in message AMTSDTI.

3. Message AMTSD2I is issued when the dump terminates, which indicates if the dump was successful or partially successful.

4. Message AMTSDBI displays the names or the address spaces that were to be dumped.
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**

```
*SETOP
  MSGID    | CAPSON    | DUMP     | ECHO     | RWAIT=nn
NOMSGID  | CAPSOFF   | NODUMP   | NOECHO   |
OUTLIM=n  | NOROLL=n  |
```

The parameters are defined as follows:

**MSGID**
This parameter indicates that message numbers are to be displayed at the beginning of messages produced by MainView SYSPROG Services.

**NOMSGID**
This parameter indicates that message numbers should not be displayed on messages produced by MainView SYSPROG Services.

**CAPSON**
This parameter indicates that messages are to be displayed in uppercase.

**CAPSOFF**
This parameter indicates that messages are to be displayed in mixed case.

**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 accessing MainView SYSPROG Services using the MainView PAS.

**NOECHO**

This parameter indicates that the commands you enter should not 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 accessing MainView SYSPROG Services using the MainView PAS.

**RWAIT=nn**

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.

**OUTLIM=n**

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.

**NOROLL=n**

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 34.

### 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:

**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.
STATUS (continued)  
- 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.

LOAD 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.

ENABLE 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).

DISABLE 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.
DELETE 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.

*exitName* 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).

*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.
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.
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
First positional parameter

LIST  This keyword displays each storage group name followed by a list of volumes managed by that storage.

The remaining keywords produce the same display as LIST, except that the keyword defines the column to be sorted.

STATUS  This keyword has been retained for compatibility and is an alias for SGNAME.

SGNAME  The data is sorted by the storage group name in ascending sequence.

VOLUME  The data is sorted by volume serial number in ascending sequence.

FREE  The data is sorted by percentage of free space in descending sequence.

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

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*
```

<table>
<thead>
<tr>
<th>AMT1S1I</th>
<th>Storage Group</th>
<th>Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMT1S2I</td>
<td>SGENG</td>
<td>SMSA13 SMSA12 SMSA01 SMSA04</td>
</tr>
<tr>
<td>AMT1S2I</td>
<td>SGSYS</td>
<td>SMS016 SMS015 SMS014 SMS013</td>
</tr>
<tr>
<td>AMT1S2I</td>
<td>SGTST</td>
<td>SMS25C HFS016 HFS002 HFS015</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT1S1I</td>
<td>SVC005 SVC004</td>
<td>SVC003 SVC002 SVC001 SVC006</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT1S1I</td>
<td>SVCDMP</td>
<td>SVC005 SVC004 SVC003 SVC002</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT1S1I</td>
<td>SGSYS</td>
<td>SVC005 SVC004 SVC003 SVC002</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT1S1I</td>
<td>SGTST</td>
<td>SVC005 SVC004 SVC003 SVC002</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMT1S1I</td>
<td>SGENG</td>
<td>SMSA13 SMSA12 SMSA01 SMSA04</td>
</tr>
<tr>
<td>AMT1S2I</td>
<td>SGSYS</td>
<td>SMS016 SMS015 SMS014 SMS013</td>
</tr>
<tr>
<td>AMT1S2I</td>
<td>SGTST</td>
<td>SMS25C HFS016 HFS002 HFS015</td>
</tr>
<tr>
<td>AMT1S0I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 2

To display information for all SMS-managed volumes that are sorted by descending percentage of free space with a VOLSER beginning with HSF, type

\texttt{sms free hfs*}

| AMTSSFI -- SG -- Volume --- Status --- -------- Space -------- Error |
|--------------------------|-----------------|-----------------|-------------------|-----------------|
| AMTSSF2 Name ID MVS SMS %Free LxTnt TotCap Flag | AMTSSF4 | AMTSCI SGTST HFS018 ONLINE ENABLE 71.59% 1,570M 2,707M |
| AMTSCI SGTST HFS006 ONLINE ENABLE 64.02% 527M 2,707M |
| AMTSCI SGTST HFS019 ONLINE ENABLE 62.54% 773M 2,707M |
| AMTSCI SGTST HFS008 ONLINE ENABLE 53.38% 700M 2,707M |
| AMTSCI SGTST HFS010 ONLINE ENABLE 49.24% 473M 2,707M |
| AMTSCI SGTST HFS00A ONLINE ENABLE 49.02% 251M 2,707M |
| AMTSCI SGTST HFS003 ONLINE ENABLE 46.70% 256M 2,707M |
| AMTSCI SGTST HFS015 ONLINE ENABLE 41.56% 759M 2,707M |
| AMTSCI SGTST HFS001 ONLINE ENABLE 39.08% 343M 2,707M |
| AMTSCI SGTST HFS016 ONLINE ENABLE 33.65% 352M 2,707M |
| AMTSCI SGTST HFS009 ONLINE ENABLE 29.26% 147M 2,707M |
| AMTSCI SGTST HFS005 ONLINE ENABLE 23.24% 122M 2,707M |
| AMTSCI SGTST HFS013 ONLINE ENABLE 22.18% 1,079M 8,120M |
| AMTSCI SGTST HFS012 ONLINE ENABLE 22.02% 240M 2,707M |
| AMTSCI SGTST HFS017 ONLINE ENABLE 20.87% 296M 2,707M |
| AMTSCI SGTST HFS007 ONLINE ENABLE 19.47% 389M 8,120M |
| AMTSCI SGTST HFS011 ONLINE ENABLE 14.96% 95,000K 2,707M |
| AMTSCI SGTST HFS004 ONLINE ENABLE 11.53% 88,000K 2,707M |
| AMTSCI SGTST HFS000 ONLINE ENABLE 10.86% 55,000K 2,707M |

AMT2SOI
AMTSS4I 19 volumes selected for display
AMT2SOI
SOFTFRR

The SOFTFRR (SOF) service

- reads and summarizes software records from the LOGSTREAM or a LOGREC data set
- 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

The parameters are defined as follows:

- **modname**: when specified, only records containing the specified *modname* in the recording parameters are selected
- **N/A**: when specified, indicates that only records with a blank module name in the recording parameters are selected
- *****: when specified, indicates that all records are selected
- **MAP**: when specified without MAP, abstract data is displayed for each record, rather than a summary report.
- **LOGSTREAM**
- **dsname**

When specified without MAP, abstract data is displayed for each record, rather than a summary report.

- **MAP**: 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.
- **seqNum**: when specified, only records containing the specified sequence number (error identification number) are selected
- **system**: when specified, only the records for the specified system are selected

The *system* operand is only supported when LOGREC is recording to the logstream.
Example 1

To summarize all software records, type

softfrr

<table>
<thead>
<tr>
<th>AMTL3NI</th>
<th>Processing logstream SYSPLEX.LOGREC.ALLRECS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTL31I</td>
<td>67 Records read; 17 Records accepted</td>
</tr>
<tr>
<td>AMTL39I</td>
<td>N/A 00003 IXGINLPA 00003 IG00131 00001</td>
</tr>
<tr>
<td>AMTL39I</td>
<td>IKJEFLC 00003 ISLOAD 00001 NUCLEUS 00001</td>
</tr>
<tr>
<td>AMTL39I</td>
<td>BBM33 00001 XDCCALL 00001 IMF 00001</td>
</tr>
<tr>
<td>AMTL39I</td>
<td>BBI 00002</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>This record was obtained from the logstream.</td>
</tr>
</tbody>
</table>
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
```

<table>
<thead>
<tr>
<th>AMTL30I Software Logrec Report (Vol=ES43OM; Dsn=SYS1.LOGREC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTL31I ErrorID: CPUID: 0000; SeqNo: 937; Time: 14.26.53.3; ASID: 008A</td>
</tr>
<tr>
<td>AMTL32I Abend S13E on 04/23/97 at 14.26.53.68 -- IKJEFT01.IKJEFTSC.IKJEFT05</td>
</tr>
<tr>
<td>AMTL34I Err-PSW: 078C0000 80077A30; ILC 2; INTC 0001; Trans: 00000000 [1]</td>
</tr>
<tr>
<td>AMTL3HI Data at PSW-6  4100,0001,0A01,D503,401C,C4E8</td>
</tr>
<tr>
<td>AMTL35I EFLGS: System issued SVC 13</td>
</tr>
<tr>
<td>AMTL35I :Cleanup only - Retry not allowed</td>
</tr>
<tr>
<td>AMTL35I :ErrorID information available</td>
</tr>
<tr>
<td>AMTL36I Err-Regs  0-3  00000001 11A07F98 00000000 00050A38</td>
</tr>
<tr>
<td>AMTL36I Err-Regs  4-7  11A07F7C 11A323E8 00000000 000C86E0</td>
</tr>
<tr>
<td>AMTL36I Err-Regs  8-11 00050A38 0010F768 00077B84 0010EF28</td>
</tr>
<tr>
<td>AMTL36I Err-Regs 12-15 800777A0 00CB3300 0077B7C 807B8658</td>
</tr>
<tr>
<td>AMTL37I RB-PSW: 071C1000 810FF542; ILC 2; INTC 000D [2]</td>
</tr>
<tr>
<td>AMTL38I RB-Regs  0-3  00000001 FF84C804 83F75728 03F76727</td>
</tr>
<tr>
<td>AMTL38I RB-Regs  4-7  00B7418 00000000 00B3E20 007FE030</td>
</tr>
<tr>
<td>AMTL38I RB-Regs  8-11 00000000 80B3D4B 007B49B8 03F77726</td>
</tr>
<tr>
<td>AMTL38I RB-Regs 12-15 007B32F0 007B32F0 007B32F0 807B76C8</td>
</tr>
<tr>
<td>AMTL3DI Recovery routine percolated error [3]</td>
</tr>
<tr>
<td>AMTL3G1 RFLGS: LOGREC recording requested</td>
</tr>
<tr>
<td>AMTL3A1 Variable Recording Area (VRA) Data [4]</td>
</tr>
<tr>
<td>AMTL3II VRACBM Control block macro name: TMPWRKA2</td>
</tr>
<tr>
<td>AMTL3II VRADA</td>
</tr>
<tr>
<td>AMTL3II VRACBA Control block address: 00005000F2F8</td>
</tr>
</tbody>
</table>
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.

```
sof,sjsf
```

```
sof,,sjsf
AMTL3NI Processing logstream SYSPLEX.LOGREC.ALLRECS
AMTL31I  08 Records read;  1 Records accepted
AMTL39I N/A      00000 IEWLR0O0 00001
```


SP2

The SPACE service that was available in previous versions of MainView SYSPROG Services has been renamed to the SP2 service. The SP2 service is retained in this version of the product for migration compatibility with MainView AutoOPERATOR and will be discontinued in the next version of MainView SYSPROG Services. The new SPACE service is described in “SPACE” on page 393.

NOTE

The SP2 service does not support the extended area of Extended Addressability Volume (EAV) devices.

The SP2 service displays the total amount of free space and the largest amount of contiguous free space on a volume in cylinders and tracks, indicates the amount of fragmentation on the volume.

The displayed values that exceed five digits are automatically scaled. For example, a value of 123,456 will be displayed as 123K. Scale suffixes of M, G, T, P, and E are used when appropriate.

Syntax

```
SP2
   SYSDA
   dvn
   volser
   generic
   partial
   nnn
   ALL
   UNIT
   VOLSER
   FREE
   CONTIG

SYSDA
   SYSDA (default)

   dvn
   is a device number

   volser
   is a complete six-character volume serial number

   generic
   is a generic or esoteric device name

   partial
   is the first one to five characters of a volume serial number
```
Example 1

To display information for the ten online volumes that contain the largest amount of contiguous free space, type

\texttt{sp2 sysallda free}

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS10I Esoteric unitname SYSALLDA defines 15280 devices AMTS1BI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS15I Unit Volume Atr Free Cyl Free Trk Cntg Cyl Cntg Trk Frag Indx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS19I 2400 EAVNL1 PRV 225K 142 154K 0 88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS19I AD03 --AD03 PRV 65,517 14 65,517 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS19I AD00 --AD00 PRV 65,517 4 65,517 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Legend:

1. GENERIC or ESOTERIC
2. requested generic or esoteric unit name
3. number of devices described by the generic or esoteric unit name
4. unit address
5. volume name
6. device type:
   - PRIVATE
   - PUBLIC
   - STORAGE

7. number of free cylinders and tracks
8. largest number of contiguous cylinders and tracks

**Example 2**

To display the unused space on all disk volumes that begin with the letters BAB and that contain over 500 cylinders of contiguous free space, type

```
sp2,bab,>500,contig
```

<table>
<thead>
<tr>
<th>Unit</th>
<th>Volume</th>
<th>Atr</th>
<th>Free Cyl</th>
<th>Free Trk</th>
<th>Cntg Cyl</th>
<th>Cntg Trk</th>
<th>Frag Indx</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS19I</td>
<td>AC09</td>
<td>BAB910</td>
<td>PRV</td>
<td>3,127</td>
<td>31</td>
<td>2,224</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>831A</td>
<td>BAB201</td>
<td>PRV</td>
<td>2,037</td>
<td>1</td>
<td>2,031</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>85C4</td>
<td>BAB904</td>
<td>PRV</td>
<td>1,929</td>
<td>23</td>
<td>1,929</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>8319</td>
<td>BAB200</td>
<td>PRV</td>
<td>1,744</td>
<td>14</td>
<td>1,744</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>8354</td>
<td>BAB907</td>
<td>PRV</td>
<td>2,826</td>
<td>0</td>
<td>1,594</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>876E</td>
<td>BAB906</td>
<td>PRV</td>
<td>2,371</td>
<td>0</td>
<td>1,475</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>8340</td>
<td>BAB902</td>
<td>PRV</td>
<td>5,432</td>
<td>276</td>
<td>1,320</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>852F</td>
<td>BAB903</td>
<td>PRV</td>
<td>2,832</td>
<td>24</td>
<td>1,298</td>
<td>2</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>AC0C</td>
<td>BAB911</td>
<td>PRV</td>
<td>2,601</td>
<td>115</td>
<td>1,225</td>
<td>11</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>876D</td>
<td>BAB905</td>
<td>PRV</td>
<td>1,385</td>
<td>32</td>
<td>1,175</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>AC0E</td>
<td>BAB913</td>
<td>PRV</td>
<td>3,039</td>
<td>154</td>
<td>900</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>839D</td>
<td>BAB908</td>
<td>PRV</td>
<td>1,074</td>
<td>15</td>
<td>769</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>AC0D</td>
<td>BAB912</td>
<td>PRV</td>
<td>1,754</td>
<td>203</td>
<td>740</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>85C2</td>
<td>BAB900</td>
<td>PRV</td>
<td>697</td>
<td>30</td>
<td>689</td>
<td>0</td>
</tr>
<tr>
<td>AMTS19I</td>
<td>833F</td>
<td>BAB901</td>
<td>PRV</td>
<td>2,718</td>
<td>488</td>
<td>675</td>
<td>9</td>
</tr>
<tr>
<td>AMT001A</td>
<td>SYSPROG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

392  MainView SYSPROG Services User Guide and Reference
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.

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

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. To assist you in transitioning to the new format, the previous version of the SPACE service is being retained for a limited time under the name SP2. See “SP2” on page 390.
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 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.

The default value is TRACK.
- 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:

\begin{align*}
\text{label} \\
\text{label operator value}
\end{align*}

- label is one of the following column header labels: EV, TOTAL, FREE, LARGEST, PCT, and FRAG.

- operator is one of the following operands: <, =, or >.

- value is the desired limit value.

The variable value can be N or Y for keyword EAV. You can choose

- EAV=N to reject all EAV volumes
- EAV=Y to reject all non-EAV volume
- 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 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>
<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>NA</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>
</tbody>
</table>
Table 6  SPACE service keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Sort order</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<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*`

---

SPACE service keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>DESCRIPTION</th>
<th>SORT ORDER</th>
<th>OPERATORS</th>
</tr>
</thead>
<tbody>
<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*`
**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
```

---

**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
```

---

```plaintext
AMTSP3I 8 devices selected
AMTS41I
AMTS42I Track-Managed:
AMTS43I Unit Volume Atr EAV Bytes Bytes Extent Free Index
AMTS44I Total Free largest Pct Frag
AMTS45I 839D BAB908 PRV N 2,944M 2,124M 1,921M 72% 44
AMTS45I 8354 BAB907 PRV N 2,944M 2,333M 1,495M 79% 134
AMTS45I 831A BAB201 PRV N 1,910M 1,486M 1,461M 77% 8
AMTS45I 8319 BAB200 PRV N 1,910M 1,331M 1,330M 69% 0
AMTS45I 85C4 BAB904 PRV N 2,944M 2,112M 1,262M 71% 119
AMTS45I 876E BAB906 PRV N 2,944M 2,079M 1,201M 70% 116
AMTS45I AC0F BAB914 PRV N 8,833M 1,397M 1,180M 15% 57
AMTS45I AC0C BAB911 PRV N 8,833M 1,859M 1,097M 20% 153
```

---

```plaintext
AMTSP3I 4 devices selected
AMTS41I
AMTS42I Track & Cyl-Managed:
AMTS43I Unit Volume Atr EAV Bytes Bytes Extent Free Index
AMTS44I Total Free largest Pct Frag
AMTS45I 2401 EAVNS1 PRV Y 60,849M 20,199T 30,116M 49% 3
AMTS45I 2200 EVANLA PRV Y 232G 12,372T 44,887M 21% 34
AMTS45I 2400 EAVNLI PRV Y 232G 32,823T 118G 72% 73
AMTS45I 2201 EVANSA PRV Y 60,849M 17,421T 6,406M 81% 188
```

---
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

```
SRM
  asi
  MAP
```

The parameters are defined as follows:

- `asi` address space instance; see “Identifying address spaces” on page 17 for more information
- `MAP` displays the addresses of SRM control blocks

Example 1

To display system resource information, type

```
srm
```

<table>
<thead>
<tr>
<th>AMTS30I</th>
<th>System is in goal mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS31I</td>
<td>Address Spaces : In=42 O/Rdy=0 O/Nrdy=29 Logwt=63</td>
</tr>
<tr>
<td>AMTS32I</td>
<td>Resource Status: CPU=8% High UIC=1800</td>
</tr>
<tr>
<td>AMTS36I</td>
<td>Paging Rate/Sec: Demand=0 Total=45 Delay Time=16,384</td>
</tr>
<tr>
<td>AMTS39I</td>
<td>Think-time(Sec): Cur=0.0 Min=0.0 Max=5.0</td>
</tr>
</tbody>
</table>
Example 2

To display the addresses of SRM control blocks as well, type

```
srm map
```

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>AMTS30I System is in goal mode</td>
<td>[1]</td>
<td>[2]</td>
<td>[3]</td>
</tr>
<tr>
<td>AMTS31I Address Spaces: In=40</td>
<td>0/Rdy=0</td>
<td>0/Nrdy=29</td>
<td>Logwt=65</td>
</tr>
<tr>
<td>AMTS32I Resource Status: CPU=2%</td>
<td>High UIC=1900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS36I Paging Rate/Sec: Demand=0</td>
<td>Total=0</td>
<td>Delay Time=22.016</td>
<td></td>
</tr>
<tr>
<td>AMTS39I Think-time(Sec): Cur=0.0</td>
<td>Min=0.0</td>
<td>Max=5.0</td>
<td></td>
</tr>
<tr>
<td>AMTS33I RMCT 0188D420</td>
<td>RMCA 0188DC90</td>
<td>RMPT 0188DC08</td>
<td>CCT 0188D708</td>
</tr>
<tr>
<td>AMTS34I ICT 0188DB20</td>
<td>MCT 0188DB68</td>
<td>LSCT 0188DE30</td>
<td>DMDT 022F8C40</td>
</tr>
<tr>
<td>AMTS35I CMCT 0188DEA8</td>
<td>CPMT 0204F3E0</td>
<td>CMB 00000000</td>
<td>CPWK 0204F3C0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS71I TSU09423 BMVDWP5</td>
<td>IKJDWP2</td>
<td>Service Class=TSONRM</td>
<td></td>
</tr>
<tr>
<td>AMTS72I Transaction data: Interval 00.00.07</td>
<td>[5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS73I Last Swap : Interval 00.00.07 Reason=&lt; IN MEMORY &gt;</td>
<td>[6]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS74I Swapping Profile: Count 1</td>
<td>[7]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS75I Service Data : CPU 1,264</td>
<td>[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS79I : SRB 29</td>
<td>[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS76I : I/O 190</td>
<td>[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS77I : MSO 0</td>
<td>[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTS78I : Total 1,483</td>
<td>[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>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.

If you type SSVT without parameters, SSCVT information from all subsystems is listed.
## Example

To map information from the SSCVT, type

```bash
ssvt
```

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTB61</td>
<td>SSCVT=005D26E8</td>
<td>NAME=JES2</td>
<td>(D1C5E2F2)</td>
<td>SSVT=005D5450</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=005D26C0</td>
<td>NAME=MSTR</td>
<td>(D4E2E3D9)</td>
<td>SSVT=00950EE8</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=005D2010</td>
<td>NAME=CICS</td>
<td>(C3C9C3E2)</td>
<td>SSVT=00000000</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=00950908</td>
<td>NAME=JES1</td>
<td>(D1C5E2F1)</td>
<td>SSVT=00000000</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=009508E0</td>
<td>NAME=OCCF</td>
<td>(D6C3C3C6)</td>
<td>SSVT=00000000</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=009508B8</td>
<td>NAME=RES.</td>
<td>(D9C5E240)</td>
<td>SSVT=00000000</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=009490F8</td>
<td>NAME=SIMS</td>
<td>(E2C9D4E2)</td>
<td>SSVT=00000000</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=00631698</td>
<td>NAME=VX..</td>
<td>(E5E7000E)</td>
<td>SSVT=00610EC8</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=006621C8</td>
<td>NAME=IMF1</td>
<td>(C9D4C6F1)</td>
<td>SSVT=006621F0</td>
</tr>
<tr>
<td>AMTB61</td>
<td>SSCVT=006648C8</td>
<td>NAME=RB20</td>
<td>(D9C2F2F0)</td>
<td>SSVT=00664938</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 SSVT for this subsystem; zero if the subsystem is not active
5. contents 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

IN displays the status of all address spaces swapped in memory (default)
ALL displays the status of all address spaces
TSO displays the status of all TSO users
asi address space instance; see “Identifying address spaces” on page 17 for more information
SHORT shortens the display to accommodate console screen width
Example

To display the status of all active address spaces swapped in memory, type

```
status
```

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTS21I</td>
<td>MASTER*(0001) NSW</td>
<td>FM(15704)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>94.44 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>PCAUTH (0002) NSW</td>
<td>FM( 2155)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>0.00 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>RASP (0003) NSW</td>
<td>FM( 834)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>0.86 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>TRACE (0004) NSW</td>
<td>FM(27730)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>0.00 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>DUMPSRV (0005) NSW</td>
<td>FM( 702)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>9.43 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>XCFAS (0006) NSW</td>
<td>FM(18134)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>386.06 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>GRS (0007) NSW</td>
<td>FM( 4764)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>50.17 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>SMSPDSE (0008) NSW</td>
<td>FM(10646)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>120.32 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>SMSVSAM (0009) NSW</td>
<td>FM(28322)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>73.95 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>CONSOLE (000A) NSW</td>
<td>FM( 5127)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>72.56 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>WLM (000B) NSW</td>
<td>FM(10493)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>182.69 Sec.</td>
</tr>
<tr>
<td>AMTS21I</td>
<td>ANTMAIN (000C) NSW</td>
<td>FM( 1442)</td>
<td>WS( *** )</td>
<td>CPU</td>
<td>0.45 Sec.</td>
</tr>
</tbody>
</table>

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

---

**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**

```plaintext
SUBMIT member dsname
```

The parameters are defined as follows:

- `member` is a member in the BBPARM library data set
- `dsname` is the name of the data set containing the member to be submitted

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
```

```
AMTV11I MEMBER BACKUP SUBMITTED TO JOB ENTRY SYSTEM
```

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
- 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 266.

Syntax

```
SVCFIND
```

```
ALL
```

```
200
```

```
255
```

```
type#
```

```
starting#
```

```
ending#
```

```
LIST
```

```
200
```

```
255
```

```
svc#
```

```
DUMP
```

The parameters are defined as follows:

- `type#` is a one-digit SVC type number (1, 2, 3, 4, or 6)
  
  SVCFIND displays unused SVC table entries that equal the type number.

- `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.
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
[1] [1] [1] [1] [2] [3] [4]
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
[1] [1] [1] [1] [2] [3] [4]
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   N  NONE 00FDB368  C8000000
```

To dump the first 64 bytes of storage for SVC 235, type

svcfind dump 235

```
AMTY15I SVC 235 DISPLAY
 [5] [6]
AMTY16I 813BA1C0 47F0601A 14C9C7C3 F0F0F340 F0F3F1F9 *.0...1G003.0319*
AMTY16I 013BA1D0 F540C8C2 C2F7F7F0 F9004120 6FFF189F *5.HBB7709...?...*
AMTY16I 013BA1E0 18C11830 41F00000 58002889 5810706C *.A...O.....i...%*
AMTY16I 013BA1F0 58101070 5810101C 1881181C 180350D0 *.........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.

2. highest lock required by the SVC first-level interrupt handler; the lock is obtained from the SVCLOCKS field of the SVCTABLE

3. address of the SVCTABLE entry for the appropriate SVC

4. hexadecimal display of the **SVCATTR** and **SVCLOCKS** fields from the SVCTABLE entry

5. starting address of the storage being dumped, and the hexadecimal and EBCDIC representation of that data

6. text representation of the dump
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 174 for information about clearing dump data sets.)

Syntax

```
SYSDUMP
```

Example

To display information about each active dump data set, type

```
sysdump
```

```
[1] [2]
AMTDU01I  SYS1.DUMP01  WAS FILLED AT 10:21 ON MAR 14, 2003
[3]
AMTDU11I  SOURCE=SVCDUMP
[4]
AMTDU21I  TITLE=ERROR IN TSO
```

Legend:

1. full-dump data set
2. time and date of dump
3. system component that took the dump
4. message text describing 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

<table>
<thead>
<tr>
<th>TCB</th>
<th>as</th>
<th>MAP</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>@TCBn</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The parameters are defined as follows:

- **asi**: address space instance; see “Identifying address spaces” on page 17 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.

- **@TCBn**: 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 @TCBnnn as the first parameter.

- **MAP**: 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 is to be displayed. However, symbolic names are not assigned to the various fields.

- **ALL**: 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.

**Example 1**

To display summary information for all of the tasks (except the first two) within the address space with jobname VAM3R7, type
TCB

<table>
<thead>
<tr>
<th>Address Space: VAM3R7 Step: 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTT72I @TCB4 TCB 6FF3A0 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>4 [3] AMTT7YI OTC 6FF130 NTC 000000 LTC 6C4528 JSTCB 6FF3A0 0.00 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6FF020 EP AF300000 WLIC 00020001 SYSPROG</td>
</tr>
<tr>
<td>AMTT7XI PSW 070C0000 AF300592 WT 01 CDE 006FFA28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address Space: VAM3R7 Step: 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTT72I @TCB5 TCB 6C4528 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>5 AMTT7YI OTC 6FF3A0 NTC 000000 LTC 6C4390 JSTCB 6FF3A0 0.00 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6FF318 EP AF301000 WLIC 00020001</td>
</tr>
<tr>
<td>AMTT7XI PSW 070C1000 AF301236 WT 01 CDE 006FC798</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address Space: VAM3R7 Step: 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTT72I @TCB6 TCB 6C4390 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>6 AMTT7YI OTC 6C4528 NTC 000000 LTC 6A0BF8 JSTCB 6FF3A0 0.04 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6C4308 EP AF307000 WLIC 0002007D ASTS$A1MN</td>
</tr>
<tr>
<td>AMTT7XI PSW 074C0000 AF380986 WT 01 CDE 006FF2DB</td>
</tr>
<tr>
<td>AMTT72I @TCB7 TCB 6A0BF8 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>7 AMTT7YI OTC 6C4390 NTC 6A0EB8 LTC 000000 JSTCB 6FF3A0 0.00 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6D72B0 EP AF39F000 WLIC 00020001 ASTS$VQI</td>
</tr>
<tr>
<td>AMTT7XI PSW 074C1000 AF39F5F8 WT 01 CDE 006C4258</td>
</tr>
<tr>
<td>AMTT72I @TCB8 TCB 6A0EB8 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>7 AMTT7YI OTC 6C4390 NTC 6C4068 LTC 000000 JSTCB 6FF3A0 0.00 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6A0AE8 EP AF39E000 WLIC 00020001 ASTS$AEW</td>
</tr>
<tr>
<td>AMTT7XI PSW 074C1000 AF39E2E8 WT 01 CDE 006C4278</td>
</tr>
<tr>
<td>AMTT72I @TCB9 TCB 6C4068 Prty 255 NDSP 0400 PK 40</td>
</tr>
<tr>
<td>AMTT7LI ------</td>
</tr>
<tr>
<td>7 AMTT7YI OTC 6C4390 NTC 6D70A8 JSTCB 6FF3A0 0.01 CPU</td>
</tr>
<tr>
<td>AMTT73I PRB 6D7338 EP AF38B000 WLIC 00020001 ASTS$AEW</td>
</tr>
</tbody>
</table>
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
To display location-related information (such as the subpool, key, and load module name if applicable) and offset into the load module, type

\texttt{where,r12}

\begin{verbatim}
AMTC1ZI 00000000_2F32C600 KEY= 0 VAM3R7  (ASID 005A)
AMTC1ZI 2F32C600 A7F40044 58C1E2E3 5BE2C4E5 D95EF0F5 *x4
AST$SDVR;05*
AMTC1ZI 2F32C610 61F0F261 F1F25EF1 F24BF4F4 5EC4C5E5
*/02/12;12.44;DEV*
AMTC1ZI 2F32C620 6DD3E5D3 5E404040 5E404040 5E404040 *_LVL: ;
AMTC1ZI 2F32C630 40404040 5E5E5E5E 2F32C6A4 5E4DC35D *...:
Fu:(C)*
\end{verbatim}

\textbf{Example 4}

To display more detailed information about a TCB, type

\texttt{tcb,@tcb5}

\begin{verbatim}
AMTT71I Address Space: XTST6PAS.PAS6 Step: PAS6
AMTT72I @TCB5 TCB 6EE220 Prty 255 NDSP 0400 PK 40
AMTT7LI---|------------------------------------------------------5
AMTT7YI  OTC 6DDAE8 NTC 6DD858 LTC 6C2E88 JSTCB 6DDAE8  0.00 CPU Sec
AMTT73I PRB  6C9278 EP AF3D06B0 WLIC 00020001 BBM9ZB01
AMTT7XI PSW 074C1000 AF3D0BF8 WT 01 CDE 006CA3C8
AMTT7QI Low GPR  0- 3  00000001  80F1610C  7F0E9E1C  006CE8B4
AMTT7QI Low GPR  4- 7  00000040  006C7F68  2F3D1F28  006CE848
AMTT7QI Low GPR  8-11 AF3D0BEC 2F3D06B0 7F0E9FB8 006C6C98
AMTT7QI Low GPR 12-15 2698B5A8 7F0E9CC0 AF3D0FEC 806C9278
AMTT7RI Access Regs 0- 3  00000000  00000000  00000000  00000000 [10]
AMTT7RI Access Regs 4- 7  00000000  00000000  00000000  00000000 [10]
AMTT7RI Access Regs 8-11  00000000  00000000  00000000  00000000 [10]
AMTT7RI Access Regs 12-15  00000000  00000000  00000000  00000C05 [10]
AMTT76I The task is Waiting on ECB1 at 006C6CF8
AMTT76I The task is Waiting on ECB2 at 7F0E9E34
\end{verbatim}
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

TIME GMT

The parameter is defined as follows:

GMT displays the Greenwich Mean Time specified by the operator at the time of IPL

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
The TIOT (TIO) service displays all allocated devices for a specified address space and monitors the usage of data sets by address space.

**Syntax**

```plaintext
TIOT asid [ MAP | ACT | ddname ]
```

The parameters are defined as follows:

- **asid**: address space instance; see “Identifying address spaces” on page 17 for more information
- **MAP**: displays the data set names associated with each TIOT entry
- **ACT**: 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.
- **ddname**: is the name of a DD statement
  
  Specifying this parameter displays the data set names associated with the specified DD name.

**NOTE**

Within a concatenation, if more than one of the data sets are allocated to a common volume, TIOT attributes all the EXCP’s 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`

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTE11I</td>
<td>JOB</td>
<td>332</td>
<td>INVENTORY</td>
<td>UPDATE1</td>
</tr>
<tr>
<td>AMTE12I</td>
<td>DD</td>
<td>STEPLIB</td>
<td>UNIT</td>
<td>151</td>
</tr>
<tr>
<td>AMTE13I</td>
<td>DSN</td>
<td>SYS3.SYPROG LOAD</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>AMTE12I</td>
<td>DSN</td>
<td>BB.CMC.CMC310 LOAD</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>AMTE12I</td>
<td>UNIT</td>
<td>250</td>
<td>VOLUME</td>
<td>PERF01</td>
</tr>
<tr>
<td>AMTE13I</td>
<td>DSN</td>
<td>BB.CMC.CMC310 LINK</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>AMTE12I</td>
<td>DD</td>
<td>SYS00002</td>
<td>UNIT</td>
<td>254</td>
</tr>
<tr>
<td>AMTE13I</td>
<td>DSN</td>
<td>SYSTCLG.PUB001</td>
<td>[10]</td>
<td></td>
</tr>
<tr>
<td>AMTE12I</td>
<td>DD</td>
<td>SYS00003</td>
<td>UNIT</td>
<td>254</td>
</tr>
<tr>
<td>AMTE13I</td>
<td>DSN</td>
<td>SYSTCLG.PUB002</td>
<td>[10]</td>
<td></td>
</tr>
</tbody>
</table>

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
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 inventry act`.

```
AMTE11I JOB  332  INVENTRY  UPDATE1   PRTY 279  PAGE 170'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 inventry,steplib`.

```
AMTE11I JOB  332  INVENTRY  UPDATE1   PRTY 279  PAGE 170'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
```
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**

```
TOD expression [asi [LOCAL | GMT]]
```

The parameters are defined as follows:

- **expression**: `expression` can be any valid expression as described in the section titled “Expressions” on page 18. It can also be an eight- or four-byte hexadecimal literal value enclosed within quotation marks and prefixed with an X.
- **LOCAL** (optional): assumes that the value is in GMT and is to be converted to local time
- **GMT** (optional): assumes that the value is in local time and is to be converted to GMT
- **asi** (optional): denotes the address space that contains the TOD value addressed by the expression

*Note:* The value is *not* converted when neither LOCAL nor GMT is specified.

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
AMT001A 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' C45B3A532692C383' in date and time format.

**tod X'C45B3A532692C383',local**

```
AMTTD1I 11:13:13.53 AM
AMTTD2I Thursday, June 18, 2009 Local time
AMT001A SYSPROG
```
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:

- `dvn` is the device number for which outstanding I/O is to be displayed
- `volser` is the volume serial number for which outstanding I/O is to be displayed

**Example 1**

To display all outstanding I/O, type

```
tpio
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTI11I</td>
<td>TSO</td>
<td>UNIT</td>
<td>021</td>
<td>IOQ 00F832A0 DRIVER-VTAM</td>
</tr>
<tr>
<td>AMTI11I</td>
<td>INVENTORY</td>
<td>UNIT</td>
<td>238</td>
<td>WORK01 IOQ 00F91300 DRIVER-EXCP</td>
</tr>
</tbody>
</table>

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

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
The TQE service displays information about Disabled Interrupt Exit (DIE) routines scheduled for execution and pending timer interrupts for an address space.

**Syntax**

The parameters are defined as follows:

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 “Expressions” on page 18 for more information

**Examples**

To display the TQE chain for all DIEs, type

```
tqe
```

<table>
<thead>
<tr>
<th>AMTQE1II</th>
<th>Timer Queue Elements, Active at 14:04:37.66</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQE2I</td>
<td>Exit</td>
</tr>
<tr>
<td>AMTQE3I</td>
<td>Expire Time</td>
</tr>
<tr>
<td>AMTQE8I</td>
<td>-----------</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.67 0000 Unknown 01E91160 Real SRM</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.69 0060 XTSTRPAS 90CD9448 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.76 0056 XTSTOPAS 925ABB20 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.76 0060 XTSTRPAS 925A1820 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.77 0026 DC$PAS 927684D0 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.77 0089 ORLMPROC 8F2F5A50 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:37.93 004E XTSTPPAS 921F24E8 Real User SetDIE</td>
</tr>
<tr>
<td>AMTQE4I</td>
<td>14:04:38.01 0001 <em>MASTER</em> 814F61A0 Real User SetDIE</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>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 Timer Queue Elements, Active at 14:08:22.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTQE2I Exit TCB</td>
</tr>
<tr>
<td>AMTQE3I Expire Time ASID Jobname Address Type Flags Creator Address</td>
</tr>
<tr>
<td>AMTQE8I ----------- ---- -------- -------- ---- ----- ------- --------</td>
</tr>
<tr>
<td>AMTQE4I 14:08:22.29 0019 JES2 8000857E Real User Stimer 007FD098</td>
</tr>
</tbody>
</table>

The TRACK (TRA) service provides a record of services invoked during the session and writes service output to a data set or SYSOUT.

**Syntax**

![Syntax Diagram]

The parameters are defined as follows:

- **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.

- **ALL**
  - initiates tracking for all user-invoked services

- **START**
  - initiates tracking for all user-invoked services
Examples

- To initiate the recording of all services and their output to the preallocated data set `myid.output`, type
  
  `track,start,myid.output`

  **NOTE**
  A TRACK command can be included in the `$INTKxx` member to specify the name of the output data set. If `$INTKxx` 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

`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
  
  `track,ALL,myid.output`

- To stop tracking and free an output data set, type
  
  `track,FREE`

- To stop tracking (leaving an output data set allocated), type
  
  `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 427 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.
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:

- `userid` TSO user ID
- `userid*` partial TSO user ID

**Examples**

To display information about all TSO users currently logged on to the system, type

```
tsulist
```

<table>
<thead>
<tr>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTT S1I USERID ASID NODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BAONIS5 0038 TCPBL142</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BAONIS4 0085 TCPBL140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BAOJAB6 0086 TCPBL234</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BAONIS2 00AD TCPBL139</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BCVPAP2 00EA TCPBL177</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BCVPAP3 00EB TCPBL178</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BA0JDB3 00ED TCPBL341</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BA0DDL2 00EF TCPBL195</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMTT S2I BCVJPH1 00F7 TCPBL204</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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>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>

Legend:

1. TSO user ID
2. ASID (in decimal format)
3. terminal name

10 users are logged on, USERMAX= 40
26 users are logged on, USERMAX= 500
26 users are logged on, USERMAX= 500
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

The parameters are defined as follows:

\[ \text{dvn} \quad \text{volser} \]

\[ \text{dvn} \quad \text{is the device number whose UCB you want to display} \]

\[ \text{volser} \quad \text{is the volume serial number you want to display} \]

Example

To display the unit control block for volume serial number BAB320, type

\[ \text{ucb bab901} \]

```
AMTU2AI DVN=833F, Device class=DASD, Type=3390, Online, allocated, 64-Bit
AMTU2BI Volser=BAB901, Status=Private, Shared
AMTU2CI Defined paths=4, Active paths=4, LCU No=152
AMTU21I UCB 833F Relocated Prefix
AMTU2DI (First field is UCBRSTEM)
AMTU25I 021D39A0 +0464 +X01D0 00040040 00000000 *........*
AMTU25I 021D39A8 +0472 +X01D8 00000000 00012BE7 *.......X*
AMTU25I 021D39B0 +0480 +X01E0 289C0CCF B00800F0 *........0*
AMTU25I 021D39B8 +0488 +X01E8 70784432 00000000 *........*
AMTU25I 021D39C0 +0496 +X01F0 01080000 00000001 *........*
AMTU21I UCB 833F Prefix
AMTU25I 00EFBD38 -0008 -X0008 00000000 01A90F80 *.....z..*
AMTU22I UCB 833F Common Section
AMTU25I 00EFBD38 -0008 -X0008 00000000 01A90F80 *.....z..*
AMTU22I UCB 833F Common Section
AMTU25I 00EFBD40 +0000 +X0000 0088FF8C B33F0000 *.h..c...*
AMTU25I 00EFBD48 +0008 +X0008 00000000 00E4C3C2 *......UCB*
AMTU25I 00EFBD50 +0016 +X0010 3030200F 00EFBD18 *........*
```
Legend:

1. specified device or volume serial number

2. interpreted information from UCB

3. UCB relocated prefix section; the relocated prefix is a data area in the portion of the UCB prefix relocated above the 16-MB line

4. UCB prefix section for the device

5. UCB common section for the device

6. UCB device-dependent section for the device

   The length of this section is dependent upon the type of device represented by the UCB.

7. UCB common extension for this device

   The length of this section is dependent upon the type of device represented by the UCB.

8. address in hexadecimal format

9. offset from the origin in decimal and hexadecimal format

10. hexadecimal representation of data (eight bytes per line)

11. 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

The parameters are defined as follows:

- \( dvn \) is the hexadecimal device number
- \( volser \) is a complete six-character volume serial number
- IN displays only the swapped-in address spaces that are using the device (default)
- ALL displays all the active address spaces that are using the device
Example

To display the active address spaces that are using device BAB329, type

using BAB329

Legend:

1. job name

2. DD name of the DD statement for the data set allocated to this device

3. data set name
The VIO service displays usage of VIO slots by job name or user ID.

**Syntax**

```
 .. figure:: syntax_vio.png

 VIO [0] [threshold]
```

The parameter is defined as follows:

`threshold`  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

**Examples**

To display the number of VIO slots used by each job name or user ID, type `vio`

```
[1] [2]
AMTVI1I VMON1  HAS  2  VIO SLOT(S) IN USE
AMTVI1I SUP6   HAS  5  VIO SLOT(S) IN USE
AMTVI1I TESTJB2 HAS 12 VIO SLOT(S) IN USE
AMTVI1I VMON8  HAS  8  VIO SLOT(S) IN USE
AMTVI1I VMON9  HAS  2  VIO SLOT(S) IN USE
```

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)
```
The VMCMD (VMC) service lets an z/OS user running as a VM guest issue CP commands through the virtual console interface.

**Syntax**

```
  VMCMD # cpcmd
```

The parameter is defined as follows:

- `cpcmd` any valid CP command

A pound sign (#) delimits multiple CP commands.

**Example**

To enter the CP command `Q NAMES`, type

```
  vmcmd q names
```

```
 AMTVM1I TOMCFCC - DSC, MVSSJSI - DSC, CDP5 - 1321, 4ESAMAP - DSC
 AMTVM1I FESAMAP - DSC, LESAMAP - DSC, 5ESAMAP - DSC, RESAMAP - DSC
 AMTVM1I CESAMAP - DSC, MVSAPD1 - DSC, 5OPER2 - 1322, HTTPD - DSC
 AMTVM1I FINGERD - DSC, SFTFD - DSC, IDENTD - DSC, 5FTPSRV - DSC
 AMTVM1I BPUTIL - DSC, TCP5VS1 - DSC, TCPIP - DSC, TROTH - DSC
 AMTVM1I 5RSCS - DSC, 5VTAM - DSC, VM5TCPIP - DSC, 5BATCH - DSC
 AMTVM1I 5RTM - DSC, XDIR - DSC, 5GCS - DSC, 5ESAWRIT - DSC
 AMTVM1I 5VTMPROP - DSC, WRKRBE - DSC, 5TCPPROP - DSC, 5ESASERV - DSC
 AMTVM1I APDCFCC - DSC, SDACFCC - DSC, CFCONSrv - DSC, CHKSTORE - DSC
 AMTVM1I 5SYMP - DSC, 5EREP - DSC, 5OPER - DSC, MVSTOM2 - DSC
 AMTVM1I MVSTOM1 - DSC, MVSSDAO - DSC, MVSSDAN - DSC, MVSSJSF - DSC
 AMTVM1I VSM - 5VTAM
 AMTVM1I 5ESAMON - L311106
```
The WARNING (WAR) service (alias AEW) provides the ability to start and stop the Exception Monitor (AEW) and display and alter the parameters for active samplers. For an explanation of the various samplers and their parameters, see Chapter 4, “Exception Monitor samplers.”

**NOTE**

The WARNING (AEW) service is not supported when MainView SYSPROG Services is executed as a TSO command.

### Syntax

The parameters are defined as follows:

- **LIST**
  - lists the Exception Monitor samplers currently active (default)

- **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 (PWSCPMxx) 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

- **STATS**
  - displays a status report detailing the CPU time used by the Exception Monitor
  
  This report is also displayed when the Exception Manager is stopped.

- **samp**
  - `samp` starts the Exception Monitor system (`samp=[parm]`)
  - `samp` starts the Exception Monitor system (`samp=[parm=value]`)

---

**WARNING**

Chapter 5 Services 439
**Example 1**

To display the active Exception Monitor samplers, type

`warn`

```
WARN
AMTTH2I 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
PWSIN100 AEW parameters obtained from member PWSCPM01
PWSSDV65 AEW Monitor Active. 5:00:24 PM, SJSD, Release: 3.6.00
PWSSDV09 AEW Initialization Complete
AMTAW1I AEW initialization completed
```
**Example 3**

To display the status report, type

```
warn,stats
```

<table>
<thead>
<tr>
<th>AMTV31I</th>
<th>Sampler</th>
<th>Calls</th>
<th>Microsec</th>
<th>Microsec</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTV32I</td>
<td>GBLS</td>
<td>184</td>
<td>5,231</td>
<td>28</td>
<td>0.24</td>
</tr>
<tr>
<td>AMTV33I</td>
<td>JSU</td>
<td>184</td>
<td>2,039K</td>
<td>11,083</td>
<td>95.06</td>
</tr>
<tr>
<td>AMTV34I</td>
<td>Admin</td>
<td>-N/A-</td>
<td>101K</td>
<td>273</td>
<td>4.68</td>
</tr>
<tr>
<td>AMTV35I</td>
<td>Total/Avg</td>
<td>368</td>
<td>2,145K</td>
<td>5,829</td>
<td>100.00</td>
</tr>
</tbody>
</table>

**Example 4**

To display all of the parameters and their associated values for the DNR sampler, type

```
warn,dnr
```

**Example 5**

To display the value associated with PENDING parameter for the MTP sampler, type

```
warn,mtp,pending
```
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 |

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:

- `expression` 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 18 for more information.
- `asi` address space instance; see “Identifying address spaces” on page 17 for more information.

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
Example 2

To locate address 0889F010, type

wh,0889F010

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`

<table>
<thead>
<tr>
<th>AMTWH4I</th>
<th>00DE9000 Found in Pagable Link Pack Area (PLPA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTWH5I</td>
<td>At IEFBR14+0 Load ADRs=00DE9000, Len=00000008</td>
</tr>
</tbody>
</table>

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:

- **expression**
  - see “Expressions” on page 18

- **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.

- **rrrrrrrrrr**
  - 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 17 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 18 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 zSeries Integrated Information Processor (zIIP) usage for each address space that has work eligible for execution on zIIP processors.

**Syntax**

There are no parameters for the ZIIP service.

**Example**

Type

\texttt{ziip}

<table>
<thead>
<tr>
<th>Adrs space</th>
<th>CPU Milli-sec</th>
<th>ZIIP Actual</th>
<th>ZIIP Eligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMFJAVA</td>
<td>2,394,969</td>
<td>47.69%</td>
<td>0.12%</td>
</tr>
<tr>
<td>CMFJAVA2</td>
<td>219,121</td>
<td>8.70%</td>
<td>0.07%</td>
</tr>
</tbody>
</table>

Located 2 address spaces with zIIP time

**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)
Customizing MainView SYSPROG Services

After you have installed MainView SYSPROG Services, you need to perform some post-installation customization to make the product operational at your site. You can customize your products in two ways: AutoCustomization or manual customization.

- AutoCustomization—allows you to perform the minimum steps required to make your product operational
- Manual customization—allows you to customize your products to best suit your needs

**NOTE**

BMC highly recommends that you use the AutoCustomization method for customizing your products.

This appendix discusses both methods of customization.

AutoCustomization ................................................................. 448
Manual customization .............................................................. 449
AutoCustomization

AutoCustomization is an interactive, online ISPF dialog provided by BMC Software to customize installed BMC Software products. AutoCustomization minimizes mistakes, propagates information for shared customization steps, allows you to browse steps before you perform them, and marks each step as it is completed. You can also bypass steps if you prefer to perform them manually.

Invoking AutoCustomization

AutoCustomization is executed online. You do not need to allocate any libraries or modify panels before invoking AutoCustomization.

To invoke AutoCustomization

1. On any ISPF panel COMMAND line, type the following command:

   TSO EX 'hilevel.BBCLIB(BBCUST)'

2. Press Enter.

3. Supply the high-level qualifier of your target libraries, as requested by AutoCustomization.

4. Press Enter.

   AutoCustomization displays the Product Customization menu.

5. Select SYSPROG Services.

   After you select one or more products, AutoCustomization presents a comprehensive list of sequentially numbered steps that you must complete before the product is operational. The number of steps varies with the product selected. Most of the steps are required, but some are optional.

   You can get help at any time during AutoCustomization by typing HELP on the COMMAND line or by pressing the Help PF key. Requesting help at the first customization panel provides an overview of AutoCustomization.

   For instructions and additional information about using AutoCustomization, refer to the OS/390 and z/OS Installer Guide.
Manual customization

Alternatively, you can customize SYSPROG Services manually. Manual customization is discussed in the MainView Customization Reference, which includes instructions for implementing product-specific functions.

To execute manual customization, you need access authority as described in the MainView Customization Reference.

If you choose to customize SYSPROG Services using the manual customization method, you also need to complete the following steps, which are unique to SYSPROG Services:

1. Obtain authorization as described in “Obtaining authorization for SYSPROG Services” on this page.

2. Create JCL for running SYSPROG Services as a batch job or started task; see “Creating a startup procedure for SYSPROG Services” on page 450.

When you complete customization, verify that your product functions are operational by using the product as described earlier in this manual.

Obtaining authorization for SYSPROG Services

If you plan to run MainView SYSPROG Services under TSO, either from the COMMAND line or from ISPF, you must perform some additional installation steps to obtain authorization.

Authorization can be obtained through the TSO Service Facility (TSF); see “Obtaining authorization through the TSO Service Facility.”

Obtaining authorization through the TSO Service Facility

The TSO Service Facility (TSF) provides a facility that permits unauthorized programs (in ISPF) to call specific APF-authorized commands and programs.

The APF-authorized commands and programs and the programs that can use the TSF interface are defined in the AUTHCMD, AUTHPGM, and AUTHTSF lists in SYS1.PARMLIB member IKJTSOxx.
Obtaining authorization for SYSPROG Services

If you are using TSF to obtain authorization for SYSPROG Services, consult Table 9 to determine what to add to the list in member IKJTSOxx.

Table 9  How TSF users obtain authorization to MainView SYSPROG Services

<table>
<thead>
<tr>
<th>To authorize MainView SYSPROG Services</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>under ISPF</td>
<td>PMGLAUTH to the AUTHTSF list</td>
</tr>
<tr>
<td>as a program</td>
<td>SYSPROG Services to the authorized</td>
</tr>
<tr>
<td></td>
<td>program list—AUTHPGM</td>
</tr>
<tr>
<td>as a command</td>
<td>SYSPROG Services to the authorized</td>
</tr>
<tr>
<td></td>
<td>command list—AUTHCMD</td>
</tr>
</tbody>
</table>

**NOTE**

If your installation uses ACF/2, you might also need to add PMGLAUTH or SYSPROG Services or both to the ACF/2 command-limiting table.

After you have installed SYSPROG Services into an authorized library, you can use the SYSPROG Services AUTHTSO service to add or delete entries from the AUTHCMD, AUTHPGM, and AUTHTSF lists. If SYSPROG Services was not previously added to the appropriate list, you will need to execute SYSPROG Services as a batch job or a started task; in this case SYSPROG Services must reside in an authorized library.

Creating a startup procedure for SYSPROG Services

If you want to invoke SYSPROG Services as a started task or batch job, you must create JCL members and save them in a cataloged procedure data set.

- For sample JCL to invoke SYSPROG Services as a started task, see “Executing MainView SYSPROG Services as a Started Task” on page 54.

- For sample JCL to invoke SYSPROG Services as a batch job, see “MainView SYSPROG Services as a batch job” on page 57.

In addition, member @@MRX050 in the BBILIB data set contains sample JCL for executing SYSPROG Services. Modify this JCL to conform to the standards of your site.
Appendix B  Previous version of the Progress service

Release 3.7 contains two versions of the PROGRESS service. The prior version provided in Release 3.6, is being maintained in Release 3.7 so that you can use the PROGRESS service for automation, and can adapt to the new version at your convenience, see “PROGRESS” on page 337.

Currently, the Release 3.6 version is the default when SYSPROG Services is operating under MainView AutoOPERATOR. While the Release 3.7 version is the default in all other environments. You may select the desired version by specifying the version number (36 or 37) as the second parameter. The documentation for the Release 3.6 version of the PROGRESS service follows.

The PROGRESS (PRO) service displays the status and current step information for a specified address space. This service helps you determine how long an address space or step has been running.

Syntax

```
PROGRESS  asi
```

The parameter is defined as follows:

```
asi  address space instance; see “Identifying address spaces” on page 17 for more information
```

Example

To display the progress of TSO user BMVDWP3, type
pro,XTSTWPAS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AMTP90I STC00636 XTSTWPAS: SYSSTC</td>
<td></td>
</tr>
<tr>
<td>AMTP9RI Reusable=NO</td>
<td></td>
</tr>
<tr>
<td>AMTP93I MSGCLASS : R MSGLEVEL : (1,1)</td>
<td></td>
</tr>
<tr>
<td>AMTP97I Program : BBM9DA00 Pgm Name :</td>
<td></td>
</tr>
<tr>
<td>AMTP91I Job Start Date : 2006.231 19 Aug 2006 :</td>
<td></td>
</tr>
<tr>
<td>AMTP92I Job Start Time : 1:45:40 Job Class : *</td>
<td></td>
</tr>
<tr>
<td>AMTP94I Step Start Time : 1:45:40 Step : XTSTWPAS (3/3)</td>
<td></td>
</tr>
<tr>
<td>AMTP95I Step TCB Time : 936.41 Sec SRB time : 101.31 Sec</td>
<td></td>
</tr>
<tr>
<td>AMTP98I Step total CPU : 17.29 Min zAAP Elig. : 0.00%</td>
<td></td>
</tr>
<tr>
<td>AMTP99I zIIP Elig. : 0.00%</td>
<td></td>
</tr>
<tr>
<td>AMTP9MI Step CPU Limit : 1,440.00 Min Pct. used : 1.20%</td>
<td></td>
</tr>
<tr>
<td>AMTP9NI zAAP CPU time : 0.00 Sec</td>
<td></td>
</tr>
<tr>
<td>AMTP90I zIIP CPU time : 0.00 Sec</td>
<td></td>
</tr>
<tr>
<td>AMTP96I Region Requested : 7,144K</td>
<td></td>
</tr>
<tr>
<td>AMTP9CI Current use &lt;16M : 584K Extended : 11,764K</td>
<td></td>
</tr>
<tr>
<td>AMTP9DI Max used &lt;16M : 988K Extended : 191M</td>
<td></td>
</tr>
<tr>
<td>AMTP9JI Common Storage : ---CSA--- ---SQA---</td>
<td></td>
</tr>
<tr>
<td>AMTP9KI Below : 23,648 0 :</td>
<td></td>
</tr>
<tr>
<td>AMTP9LI Above : 3,788K 2,312 :</td>
<td></td>
</tr>
<tr>
<td>AMTP9G1 MEMLIMIT (LVMO) : NOLIMIT set by :</td>
<td></td>
</tr>
<tr>
<td>AMTP9HI Current : OM Hidden : OM</td>
<td></td>
</tr>
<tr>
<td>AMTP9II High-Water Mark : OM # Objects : 0</td>
<td></td>
</tr>
<tr>
<td>AMTP9EI LNKLST Set Name : IPL</td>
<td></td>
</tr>
<tr>
<td>AMT001A SYSPROG</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. these three messages form a matrix that displays 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. LNKLIST Set Name currently in use for the specified address space
Index

Symbols

$INEXCL
   definition 30
   syntax rules 31
$INLGxx 269
$INSYSO
   definition 30
   parameters 34
   syntax rules 34
$INTKxx 428
$INWTO
   definition 30
   syntax rules 33
$RPJOBS
   definition 30
   syntax rules 37
$RPTEXT
   definition 31
   syntax rules 38
$XENQ
   definition 31
   syntax rules 39
$XRES
   definition 31
   syntax rules 40
@ symbol
   defined as special character 19
   for SYSPROG services commands 228
   valid CMDID character 53

Numerics

64-bit addresses, facilitating their use 21

A

accessing SYSPROG Services
   from an operator console 53
   from easy menus 47
   from EZMFPROG 48
   from EZMSPROG 47
   from SYSPROG Services Menu panel 49
   from z/OS and USS Solutions panel 45
   with a CMDID character 53
   with the MODIFY command 53

ACPU sampler 62
   excluding job names from exception reporting 31
ACTIVATE, LNKLST function 264
active task queue, monitoring 89
address expression, using symbolic name in place of 221
address space
   frames allocated to 364
   hung 242
   ID, displaying 431
   identifying malfunctions 301
   listing regions 336
   making nonswappable 192
   making swappable 310
   modules loaded in 171
   pending timer interrupts 425
   private storage allocated 346
   private, subpool allocations 331
   queues maintained in ASVT 164
   quiescing 223
   reusing 75
   slots available for starting new address spaces 112,
   113
   summarizing virtual storage allocations 331
   SVC dump 370
   target 20
   terminating 231
   terminating tasks 185
   TSO. See TSO address spaces, monitoring
   wait limit 272
   waiting for SPOOL space 130
   warning message 130
   with above-the-bar memory objects 254
address space, displaying
   allocated devices 417
   CPU time limits 257
   device usage 435
   frame allocation 364
   Large Virtual Memory Objects 312
   LSQA usage 332, 451
   request blocks (RB) 411
   shared memory objects 312
   status 403
   step information 272, 337
   storage allocations by subpool and key 333
   task control blocks (TCB) 411
   tasks 185
TQE 425
address space, monitoring
activity 280, 295
CPU usage 62, 87
EXCPs issued 66
I/O usage 64
number in use 131
paging activity 67, 69
real-storage usage 126
SRM service components 71, 73
status 403
usage of data sets 417
vector table slots 112, 113
virtual storage use 331
Address Space CPU (ACPU) sampler 62
Address Space I/O (AIO) sampler 64
Address Space I/O Rate (AIOR) sampler 66
Address Space Paging (APAG) sampler 67
Address Space Paging Rate (APGR) sampler 69
Address Space Service Rate Percentage (ASRV) sampler 71
Address Space Service Unit Rate (ASSR) sampler 73
Advanced Early Warning (AEW) Exception Monitor samplers. See Exception Monitor samplers
AEW service 151
AIO sampler 64
excluding job names from exception reporting 31
AIOR sampler 66
excluding job names from exception reporting 31
APAG sampler 67
excluding job names from exception reporting 31
APF authorization 165
APF list
adding entries 155
changing the volser of existing entries 156
updating 153
APF service 153
APGR sampler 69
excluding job names from exception reporting 31
ASID, nonreusable 164
ASM service 157
ASRV sampler 71
excluding job names from exception reporting 31
ASSR sampler 73
excluding job names from exception reporting 31
asterisk, used in an expression 19, 20
ASVT sampler 75
ASVT service 164
AUTHCMD (CMD) authorization list 165
authorization for SYSPROG Services
through SYSPROG Services 450
trough TSO Service Facility 449
authorization lists
AUTHCMD 165
AUTHPGM 165
AUTHTSF 165
building 167
modifying 165
restoring 167
AUTHPGM (PGM) authorization list 165
AUTHTSF (TSF) authorization list 165
AUTHTSO service 165
AutoCustomization 448
AutoOPERATOR. See MainView AutoOPERATOR
auxiliary storage, managing 157
Available Queue, monitoring 75
available space on DASD volumes, monitoring 128

B
batch address spaces, monitoring
CPU usage 77
I/O usage 78
paging activity 80
SRM component usage 82
Batch CPU (BCPU) sampler 77
Batch I/O (BIO) sampler 78
Batch Job mode 27
batch job, suppressing WTOR message 33
Batch Paging (BPAG) sampler 80
Batch Service (BSRV) sampler 82
BBILIB library member, @@MRX050 450
BBPARM library member
$$INEXCL 31
$$INNWTO 33
$$INSYS0 34
$$INTKxx 428
$$RPJOBS 37
$$RPTTEXT 38
$$XENQ 39
$$XRES 40
executing commands listed in 228
list 30
PWSCPMxx 40
PWSCPMxx, setting report and sample intervals 105
BBSAMP library member, SYSPROGJ 55
BBXS service, displaying status 169
BCPU sampler 77
excluding job names from exception reporting 31
BIO sampler 78
excluding job names from exception reporting 31
block size, displaying for a specified data set 193
BMC Software, contacting 2
BMC Software, subsystem service, displaying status 169
BPAG sampler 80
excluding job names from exception reporting 31
BSRV sampler 82
excluding job names from exception reporting 31
buffers, console, monitoring 143

C
CADP sampler 84
CDE service 171
CHA sampler 85
Channel (CHA) sampler 85
channel paths, monitoring
  CADP 84
  CHA 85
  MIO 290
  MONITOR 295
  MPATH 297
character string, displaying location of occurrences 284
CLEAR service 174
CMDID characters, using to access SYSPROG Services 53
CNSL service 176
comma
  separating service name and parameters 24
  used as placeholder 24
  used in control statements 94
COMMAND service 178
commands
  Execute 52
  invoking standard sequences 228
  Locate 50
  MODIFY (F), used to access SYSPROG Services 53
  Select 51
  syntax 22, 23
Common Area Dataspace (CAD) Percentage sampler 84
Common Service Area. See CSA storage
common storage
  frames allocated to 364
  freeing 232
  monitoring usage by job name 93
comparison string, specifying start and length 22
conflicts, enqueue 217, 352
console buffers, monitoring 143, 144
console reply buffers, number in use 179
console, nonoperational 179
CONSOLES service 179
Contents Directory Entry (CDE), listing for address spaces 171
contiguous space, available, determining 390
control blocks
  ASVT 164
  LPA active queue 268
  SRM 398
  Subsystem Vector Table (SSVT) 401
control statement member, creating 61
conventions, document 16
CP commands, entering 438
CPU active task queue, monitoring 89
CPU busy percentage 342
CPU model number 240
CPU Queue Depth (CQD) sampler 89
CPU queue depth, monitoring 89
CPU sampler
  description 87
  excluding job names from exception reporting 31
  CPU serial number 240
CPU service 180
CPU time limits 257
CPU usage
  displaying by job name 180
  monitoring address spaces 62
  monitoring batch address spaces 77
  monitoring by system 87
  monitoring for entire Plex 146
  monitoring host partition 118
  monitoring TSO address spaces 133
  warning message 62, 118, 146
CQD sampler 89
CSA and SQA Utilization (CSA) sampler 90
CSA sampler 90
CSA service 182
CSA storage
  displaying usage 182
  monitoring usage 90
  monitoring usage by job name 93
CSMJ sampler 93
CSMON 185
CSNJ sampler
  excluding job names from exception reporting 31
CTCB service 185
current date, displaying 416
customer support 3
customization
  AutoCustomization 448
  manual 449
D
DASD Free Space (SPA) sampler 128
DASD I/O Queue (DIOQ) sampler 97
DASD Reserves (RES) sampler 124
DASD volumes
  determining cause of poor performance 362
  displaying data blocks 203
  displaying information about mount requests 301
  evaluating contention 293
  not ready 99
  under control of SMS 381
DASD volumes, displaying
  available space 390, 393
  data blocks by cylinder, track, and record number 205
  data blocks by data set name 206
  data blocks using the DVIEW service 203
  data set names 207
  extent list 207
  Format 4 DSCB 207
  label data block 208
  pending mount requests 301
  reserve activity 362
  VTOC entry 208
DASD volumes, monitoring
  available space 128
I/O queue depth 97
pending mount requests 111
reserve activity 124
using DNR sampler 99
DASD-mount requests, monitoring 111
data blocks, displaying
    by cylinder, track, and record number 205
    by data set name 206
    from any online DASD volume 203
data sets
    displaying information about 193
    extents 203
    usage by address space 417
data sets, system dumps
    clearing 174
    displaying title, date, and time 410
    SDUMP service 370
date, displaying 415
DCPU service 342
DD names, listing members within a concatenation 417
DEACT, LNKLST function 265
default values, using commas as placeholders 24
delay, workload, monitoring 139
DEV sampler 95
DEVIATN service 188
Device Busy (DEV) sampler 95
Device Not Ready (DNR) sampler 99
devices
    allocated to an address space 417
    detecting potential performance delays 242
    identifying excessive contention 95
    monitoring 95, 280, 295
    submitting 405
    type, displaying information about 193, 433
    used by address space 435
DFLIST service 190
diagnosing problems using the BBXS service 169
DIOQ sampler 97
Disabled Interrupt Exit (DIE) routines 425
DNR sampler 99
documentation
    conventions 16
    electronic, online Help 16
    online 16
domains, displaying processor usage 342
DONTSWAP service
    description 192
    reversing effects using OKSWAP 310
DSNAME service 193
DSSUM service 195
dump data sets
    clearing 174
    displaying title, date, and time 410
    SDUMP 370
DUMP service
    and the Last-Referenced Storage Pointer (LRSP) 20
    description 198
DVIEW service 203
DVTN sampler 100

E
easy menus
    EZMFPROG 48
    EZMSPROG 47
    EZMZOS 47
ECSA storage
    displaying usage 182
    monitoring usage by job name 93
EDTINFO service 209
elapsed time, monitoring jobs 106
electronic documentation, online help 16
Eligible Device Table (EDT) 209
ENABLE command, loading exit routine into storage 380
END OF SERVICE message 36
ENQ sampler
    description 102
    excluding job names from exception reporting 31
    excluding names from warning messages 39
Enqueue Conflicts (ENQ) sampler. See ENQ sampler
enqueues
    conflicts, monitoring 102
    displaying conflicts 217
    excluding names from display 39, 40
ENQUEUES service 217
enqueues, displaying conflicts 352
entering commands 23
EQUATE service
    description 221
    symbol names for memory objects 20
ESCLASS service 223
esoteric unit names 209
ESQA storage, monitoring usage by jobname 93
Exception Monitor
    accessing 61
    controlling 439
    excluding job names from 31
    generating AO Alerts 28
    messages, controlling 34
    setting the report interval 105
    setting the sample interval 105
    storing as members in BBPARM 40
Exception Monitor samplers
    ACPIU 62
    AIO 64
    AIOR 66
    APAG 67
    APGR 69
    ASRV 71
    ASSR 73
    ASVT 75
    BCPU 77
    BIO 78
BPAG 80
BSRV 82
CADP 84
CHA 85
CPU 87
CQD 89
CSA 90
CSMJ 93
definition 28
DEV 95
DIOQ 97
DNR 99
DVTN 100
ENQ 102
FIX 103
INT 105
JET 106
JSU 108
JVR 110
MTP 111
NRQ 112
NRQP 113
OUT 115
PAG 117
PCPU 118
PER 120
PGQ 122
REP 123
RES 124
RSU 126
SMF 127
SPA 128
SPLW 130
SRM 131
SWP 132
TCPU 133
TIO 134
TPAG 136
TSRV 137
WDLY 139
WOBJ 141
WRT 142
WTOR 144
XCPU 146

EXCPs, monitoring address spaces 66
EXECUTE service 228
executing commands 228
exempting from reply scan 37
exit point 376
exit routines 376
EXIT service 231
expressions 18
Extended Common Service Area. See ECSA storage
Extended SVC Router (ESR) table 268
extended system link-pack area. See LPA (link-pack area)
Extended System Queue Area. See ESQA storage

F
FCOMMON service 232
FINDMBR service 233
FIX sampler 103
Fixed Pages (FIX) sampler 103
fixed pages, monitoring 103
frame allocation
  displaying for an address space 364
  monitoring usage by job 108
free space
  fragmented 331
  on DASD volumes, monitoring 128
  on volume 390

G
G scale suffix 21
generic unit names 209
Gi scale suffix 21
Gigabytes scale suffix
  G 21
  Gi 21
Greenwich Mean Time (GMT), displaying 415
Gregorian time format, displaying 415

H
Help
  AutoCustomization 448
  online 16
HELP service 236
hexadecimal format, displaying storage contents 198
hexadecimal string, displaying location of occurrences 284

I
I/O bottlenecks, detecting 290
I/O interrupt activity, displaying distribution among processors 323
I/O queue depth
  monitoring DASD devices 97
  monitoring page data sets 122
I/O service 242
I/O system, monitoring
  performance 290
  warning message 64
I/O usage
  displaying 242
  monitoring address spaces 64, 280
  monitoring batch address spaces 78
  monitoring devices 280

Index 459
monitoring TSO address spaces 134
I/O, outstanding 422
IFA service 239
inactive jobs that should be active 188
indirection indicators 19
INFO service 240
Initialization Parameter Area (IPA) 245
INT service 105
interface, ISPF, online Help 16
Interval (INT) sampler 105
IO service 242
IOS driver codes 243, 423
IPL, displaying information about 240
IPLDATA service 245
ISMF service 249
ISPF EDIT 217
ISPF interface Help 16
J
JCL, submitting for operations jobs 405
JES commands, invoking standard sequences 228
JET sampler

description 106
excluding job names from exception reporting 31
JITOKEN service 253
Job Awaiting V=R Region (JVR) sampler 110
Job Elapsed Time (JET) sampler 106
job entry subsystem, submitting jobs to 405
Job Storage Usage—Real & Auxiliary (JSU) sampler 108
jobs
activity, monitoring 100, 180
awaiting V=R region 110
displaying limits 257
elapsed time 106
frame usage 108
inactive, should be active 188
monitoring elapsed time 106
monopolizing devices and channel paths 290
overuse of real and auxiliary storage 108
resuming after quiescing 223
slot usage 108
submitting JCL 405
swapped out of memory 115
use of VIO slots 437
warning when job is not active 100
with outstanding WTORs 37
Jobs Swapped Out (OUT) sampler 115
JSTORM service 254
JSU sampler

description 108
excluding job names from exception reporting 31
Julian time format, displaying 415
JVR sampler

description 110
excluding job names from exception reporting 31
K
K scale suffix 21
Ki scale suffix 21
Kilobytes scale suffix

K 21
Ki 21
L
LABEL service 256
Large Virtual Memory Objects 312
Last-Referenced Storage Pointer (LRSP) 20
LCPU service 257
line number, displaying 431
list of services 49
LIST, LNKLST function 261
LNKLST service 259
LNKLST sets 259
LOADLPA service 266
LOCATE, LNKLST function 265
LOGGING service 269
logging system 269
logical control units, monitoring 293
logical partitions (LPARs), displaying processor usage 342
logical record length, displaying for a specified data set 193
LOGREC data sets, reading software records 385
LOGSTREAM, reading software records from 385
LPA (link-pack area)

active queue 268
adding modules or SVCs
LRSP (Last-Referenced Storage Pointer) 20
LWAIT service 272
LX service 275
M
M scale suffix 21
main memory, modifying 444
main storage 444
MainView AutoOPERATOR

monitors and responds to warning messages 28
SYSPROG Services available from 149
major name (Q name) 217
managing auxiliary storage, using ASM 157
manual customization for SYSPROG Services 449
master console, issuing commands from a TSO terminal 279
MCOMMAND service 279
MDEVICE service 280
Megabytes scale suffix

M 21
Mi 21
memory locations
displaying 198
searching for strings 284
memory object
  associated with an address space 313
  searching for strings 306
MEMSCAN service 284
messages
  backlog, displayed on system console 143
  how displayed 35
  identification numbers 36
  to operator, outstanding 123
  undisplayed, per system console 179
  warning about exceeded threshold 61
  warning, suppressing. See names of individual
  samplers, suppress warning message
Mi scale suffix 21
minor name (R name) 217
MIO service 290
MLCU service 293
modes, operation 27
MODIFY (F) command, accessing SYSPROG Services 53
modules
  adding in LPA 266
  deleting 266
  displaying LOGREC data 385
  loaded in an address space 171
MONITOR service 295
monitoring
  enqueue conflicts 102
  thresholds 61
mount requests 111, 301
Mounts Pending (MTP) sampler 111
MPATH service 297
MSTORAGE service 299
MTP sampler 111
MTP service 301
Multiple Domain Facility (MDF) complex 342, 352
MVS commands, invoking standard sequences 228
MVS support 15

N
Name/Token pair information 302
Non Reusable Queue (NRQ) sampler 112
Non Reusable Queue Percentage (NRQP) sampler 113
Non-reusable Replacement Queue, monitoring 75
nonswapable address space 192
non-TP I/O, displaying 242
non-VIO slots 157
NRQ sampler 112
NRQP sampler 113
NTP service 302

O
OFIND service 306
OKSWAP service 310
OLIST service
  creating symbolic names before using OFIND 306
  description 312
  symbol names for memory objects 20
online documentation 16
online Help 16
operator system, displaying information about 240
operation
  modes 27
  parameters 318
operator commands, simulating from TSO terminal 178
operator console, simulating 176
operator messages, outstanding 123
option settings 374
OUT sampler
  description 115
  excluding job names from exception reporting 31
outstanding I/O 422
outstanding replies
  displaying 123, 360
  monitoring 123

P
P scale suffix 21
PAG sampler 117
Page Data Set I/O Queue (PGQ) sampler 122
page data sets
  monitoring I/O queue depth 122
  using ASM service 157
page fixing, monitoring 103
page reclaim rate 317
paging activity, analyzing 317
paging activity, monitoring
  address spaces 67, 69
  batch address spaces 80
  rate 117
  TSO address spaces 136
  warning message 67
Paging Rate (PAG) sampler 117
PAGING service 317
parameters
  defining run-time 34
  in syntax diagrams 23
  operation 318
  system initialization 318
PARMLIST
  service 318
PAS operating mode 27
path definition 433
path failures, detecting 297
PCPU sampler 118
Q

Q name. See major name

queues
- ASVT control block 164
- ASVT sampler 75
- monitoring 75

R

R name. See minor name

RBs, displaying internal values 411

real storage
- displaying allocation 364
- monitoring status 126

Real Storage Usage (RSU) sampler 126

record format, displaying for a specified data set 193

record length, logical, displaying for a specified data set 193

Recovery/Termination Manager 231, 315, 347

related documentation 16

related publications 15

relative share 182

REP sampler 123
- excluding job names from exception reporting 31
- replacement address 19

Replies Outstanding (REP) sampler 123

REPLIES service 360

replies, monitoring 123

reply buffers
- console, monitoring 144
- number in use 179

reporting base interval, changing 105

request blocks, displaying internal values 411

RES sampler 124
- excluding job names from exception reporting 31

reserve activity
- displaying for DASD devices 362
- monitoring DASD devices 124

RESERVES service 362

resource conflict information, displaying 218

resource ownership information, displaying 219

response time, workload, monitoring 142

RSM service 364

RSU 31

RSU sampler 126
- excluding job names from exception reporting 31

run-time parameters, defining 34

RWAIT parameter

$INSYS0 36

SETOP option 375

S

samplers. See Exception Monitor samplers

scale suffixes
- G 21
- Gi 21
- K 21
- Ki 21
- M 21
- Mi 21
- P 21
- Pi 21
- scaling decimal numbers 21
- T 21
- Ti 21

scanning memory and storage areas 284
A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z

SCT service 369
SDUMP service 370
search string, specifying 22
service classes, changing, displaying, listing 223
service objectives, workload, monitoring 141
session information, displaying 431
session, displaying type used 431
Set Counter (SCT). See SCT service
SETOP service 374
settings, option 374
SLIP trap, PER sampler 120
slots
  monitoring number 75
  monitoring usage by job 108
SMF data sets
  displaying information about 249
  full 127
SMF exits
  deleting 378
  enabling 377
  routines 376
SMF recording, not active 127
SMF sampler 127
SMFEXIT service 376
SMS (system-managed storage), displaying DASD information 381
SMSLIST service 381
SOFTFRR service 385
software errors, displaying 385
software LOGRECs, summarizing 385
software problems, diagnosing using the BBXS service 169
SP2 (formerly SPACE) service 390
SPA sampler 128
SPACE service 393
SPLW sampler 130
SPOOL space, warning message 130
SQA storage
  displaying usage 182
  frames allocated to 364
  monitoring usage 90, 93
SRB mode 202
SRM component usage
  monitoring address spaces 71, 73
  monitoring batch address spaces 82
  monitoring TSO address spaces 137
SRM sampler 131
SRM service
  components, monitoring 71, 73
  description 398
SSCVT (Subsystem Communications Vector Table) 401
SSVT (Subsystem Vector Table) 401
SSVT service 401
Start/SASI Queue, monitoring 75
Started Task mode 27, 55
Started Task, suppressing WTOR message 33
startup procedure, creating 450
Status of SMF Recording (SMF) sampler 127
STATUS service 403
storage
  displaying contents in hexadecimal and character formats 198
  group names 381
  main 444
  private. See private storage
storage areas
  scanning for character or hexadecimal strings 284
  subject address 442
storage-constrained systems, tuning 331
string occurrences, creating symbol for 284
subject address storage area 442
SUBMIT service 405
subpool allocations within a private address space 331
Subsystem Communications Vector Table (SSCVT) 401
Subsystem Vector Table (SSVT) 401
subsystem, adding without having to re-IPL 401
suffix characters, used to scale decimal numbers 21
support, customer 3
suppress warning message
  batch address space paging too much 80
  batch address space using too much CPU 77
  batch address space using too much system I/O 78
  batch address space using too much system SRM 82
  CPU usage 62
  elapsed time of jobs 106
  enqueue conflicts for specific address spaces 102
  excessive SRM service component resources used 71
  excessive SRM service components used 73
  I/O activity 64
  jobs awaiting V=R region 110
  jobs swapped out of memory 115
  outstanding operator messages 123
  overuse of real and auxiliary storage 108
  TSO address space 133, 134, 136, 137
  suppressing WTOR message 33
SVC dump
  clearing 174
  displaying date and title 410
  taking 370
SVC table
  planning installation 407
  scanning 407
  updating 268
SVCFIND service 407
Swap Rate (SWP) sampler 132
swap-out rate
  monitoring 132
  OUT sampler 115
SWP sampler 132
symbols
  attributes 20
  creating for string occurrences 284
  defining, redefining, and displaying using EQUATE 221
  overview 19
predefined 20
used in place of address expressions 221
user-created 200
syntax diagrams, how to read 22, 23
syntax shortcuts to facilitate entering 64-bit addresses 21
SYSLPARMLIB members
  CONFIGxx 100, 188
  IEASYSO0 131
  IKJTSOxx 165
  listing contents 318
SYSDUMP service 410
SYSEVENT 126
SYSPROG Easy Menu 46
SYSPROG services
  accessing 44
  AEW 151
  APF 153
  ASM 157
  ASVT 164
  AUTHTSO 165
  BBXS 169
  CDE 171
  CLEAR 174
  CNSL 176
  COMMAND 178
  CONSOLES 179
  CPU 180
  creating a startup procedure 450
  CSA 182
  CSMON 185
  CTCB 185
  definition 28
  DEVIATN 188
  DFLIST 190
  DONTSWAP 192
  DSNAME 193
  DSSUM 195
  DUMP 198
  DVIEW 203
  EDTINFO 209
  ENQUEUES 217
  entering commands 23
  EQUATE 221
  ESCLASS 223
  EXECUTE 228
  EXIT 231
  FCOMMON 232
  FINDMBR 233
  HELP 236
  IFA 239
  independent operations 54
  INFO 240
  invoking as a batch job 57, 450
  invoking as a started task 450
  invoking command sequences 228
  IO 242
  IPLDATA 245
  ISMF 249
  JITOKEN 253
  JSTORM 254
  LABEL 256
  LCPU 257
  listing all 236
  LNKLST 259
  LOADLPA 266
  LOGGING 269
  LWAIT 272
  LX 275
  MCOMMAND 279
  MDEVICE 280
  MEMSCAN 284
  MIO 290
  MLCU 293
  MONITOR 295
  MPATH 297
  MSTORAGE 299
  MTP 301
  NTP 302
  obtaining authorization through TSO service facility 449
  OFIND 306
  OKSWAP 310
  OLIST 312
  PAGING 317
  PARMLIST 318
  PFKEY 321
  PIO 323
  POST 325
  PPT 326
  PRIVATE 331
  PROGRESS 337
  PRSM 342, 352
  PSTORAGE 346
  QUEUE 352
  REPLIES 360
  RESERVES 362
  RSM 364
  SCT 369
  SDUMP 370
  SETOP 374
  SMFEXIT 376
  SMSLIST 381
  SOFTFRR 385
  SP2 (formerly SPACE) 390
  SPACE 393
  SRM 398
  SSVT 401
  Started Task (STC) 54
  starting 55
  STATUS 403
  SUBMIT 405
  SVCFIND 407
  SYSDUMP 410
Terminal I/O Task service 417
terminal name, displaying 431
terminating
  address space 231
  tasks in an address space 185
thresholds
  monitoring 61
Ti scale suffix 21
TIME service 415
time, displaying 415
Timer Queue Elements (TQE) 425
TIO sampler 134
  excluding job names from exception reporting 31
TIOT service 417
TOD service 420
TP lines, enabled 422
TPAG sampler 136
  excluding job names from exception reporting 31
TPIO service 422
TQE service 425
TRACK service 427
tracks, displaying for a specified data set 193
TSF. See TSO Service Facility
TSO
  See also authorization lists
  Line mode 56
  operating mode 27
  session information 431
TSO address spaces, monitoring
  CPU usage 133
  I/O usage 134
  paging activity 136
  SRM component usage 137
TSO CPU (TCPU) sampler 133
TSO I/O (TIO) sampler 134
TSO Paging (TPAG) sampler 136
TSO Service (TSRV) sampler 137
TSO Service Facility 449
TSRV sampler 137
  excluding job names from exception reporting 31
TSULIST service 431
typographical conventions 16

U
UCB service 433
UNDEFINE, LNKLST function 265
underscore, used in 64-bit addresses 21
undisplayed messages per console services
  CONSOLES 179
unit control blocks (UCB), displaying 433
unit names, displaying information about 209
Unreferenced Interval Count (UIC) 131
UPDATE, LNKLST function 264
user ID
  displaying 431
use of VIO slots 437
USERS, LNKLIST function 262
USING service 435

V
V=R region, monitoring waiting jobs 110
VIO service 437
VIO slots
ASM service 157
displaying usage by job name or user ID 437
virtual console interface 438
virtual storage areas
allocating for private storage 331
displaying usage 299
mapping 299
VM user, issuing CP commands through virtual console interface 438
VMCMD service 438
volser
changing 153
displaying for a specified data set 193
volume serial number
changing 153
displaying for a specified data set 193
volume serial status 433
volumes
available contiguous space 390
DASD, free space (SPA) sampler 128
VSAM data sets 194
VTOC directory, searching 203

W
wait limit for an address space 272
wait state 179, 272
warning 93
warning message
address space issuing EXCPs at high rate 66
address space using too much CPU time 62
address space using too much system I/O 64
address spaces in use 131
available address space vector table slots (ASIDs) 112, 113
available SPOOL space 130
average queue depth exceeding threshold 89
batch address space paging too much 80
batch address space using too much CPU 77
batch address space using too much system I/O 78
batch address space using too much system SRM 82
channel path exceeded threshold 85
console buffer usage 143
console reply buffer usage 144
CPU utilization for host partition 118
CSA or SQA usage exceeding threshold 90
DASD devices online but not ready 99
depth of I/Os queued to page data sets 122
elapsed time of jobs 106
enqueue conflicts 102
excessive device contention 96
excessive number of I/Os queued to DASD device 97
excessive paging during reporting interval 69
excessive rate of SRM service components 73
excessive SRM service component resources used 71
fixed pages 103
free space on DASD volumes 128
job is not active 100
jobs awaiting V=R region 110
jobs meeting or exceeding thresholds 93
jobs swapped out of memory 115
outstanding operator messages 123
overuse of real and auxiliary storage 108
page fixing 103
paging activity 67
real storage use excessive 126
reserved status of devices in shared DASD system 124
slot number low 75
SMF data, potential loss 127
suppressing. See suppress warning message
system paging rate 117
system swap-out rate 132
tape and DASD-mount requests 111
too much paging activity 67
TSO address space 133, 134, 136, 137
workload delay 139
workload response time 142
workload service objectives 141
WARNING service 61, 439
WDLY sampler
excluding job names from exception reporting 31
WDLY service 139
WHERE service 442
WOBJ sampler 141
workload
monitoring delays 139, 290
monitoring response time 142
monitoring service objectives 141
Workload Delay sampler 139
Workload Objective (WOBJ) sampler 141
Workload Response Time (WRT) sampler 142
WRT sampler 142
WTU Buffer Usage (WTO) sampler 143
WTO sampler
description 143
WTOR messages
containing specific text 38
displaying 360
WTOR sampler
description 144
suppressing messages using $INNWTO 33
WTORs, exempting from reply scan 38

**X**

XCPU sampler 146

**Z**

*z/OS and USS Solutions panel, accessing SYSPROG Services from* 45  
z/OS easy menu 47  
z/OS support 15  
zAAP processors 239, 446, 453  
ZAP service  
and the Last-Referenced Storage Pointer (LRSP) 20  
description 444  
ZIIP service 446  
zSeries Application Assist Processor (zAAP). See zAAP processors  
zSeries Integrated Information Processor (zIIP) usage 446